



BYOP: Custom Processor Development with Apache NiFi

Andy LoPresto | [@yolopey](https://twitter.com/yolopey)

Cloudera Data in Motion Security

Dataworks Summit Barcelona | March 21, 2019

Agenda

- Apache NiFi out of the box
- Extending NiFi functionality
- Custom processor development
- Testing
- Deploying
- Best practices
- Takeaways
- Additional resources
- Questions

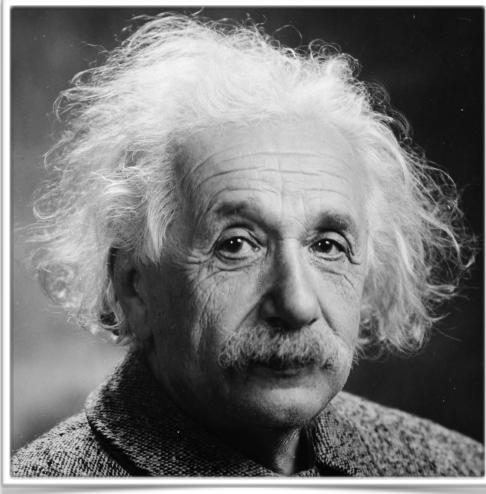
* All slides provided online <https://github.com/alopresto/slides>

Gauging Audience Familiarity With NiFi



"What's a NeeFee?"

No experience with dataflow
No experience with NiFi



"I can pick this up pretty quickly"

Some experience with dataflow
Some experience with NiFi



"I refactored the Ambari integration endpoint to allow for mutual authentication TLS during my coffee break"

Forgotten more about NiFi than most of us will ever know

Apache NiFi

One Minute Intro to the NiFi Ecosystem



NiFi

- Server/DC class
- GUI & REST API
- Interacts with hundreds of services



MiNiFi (JVM/C++)

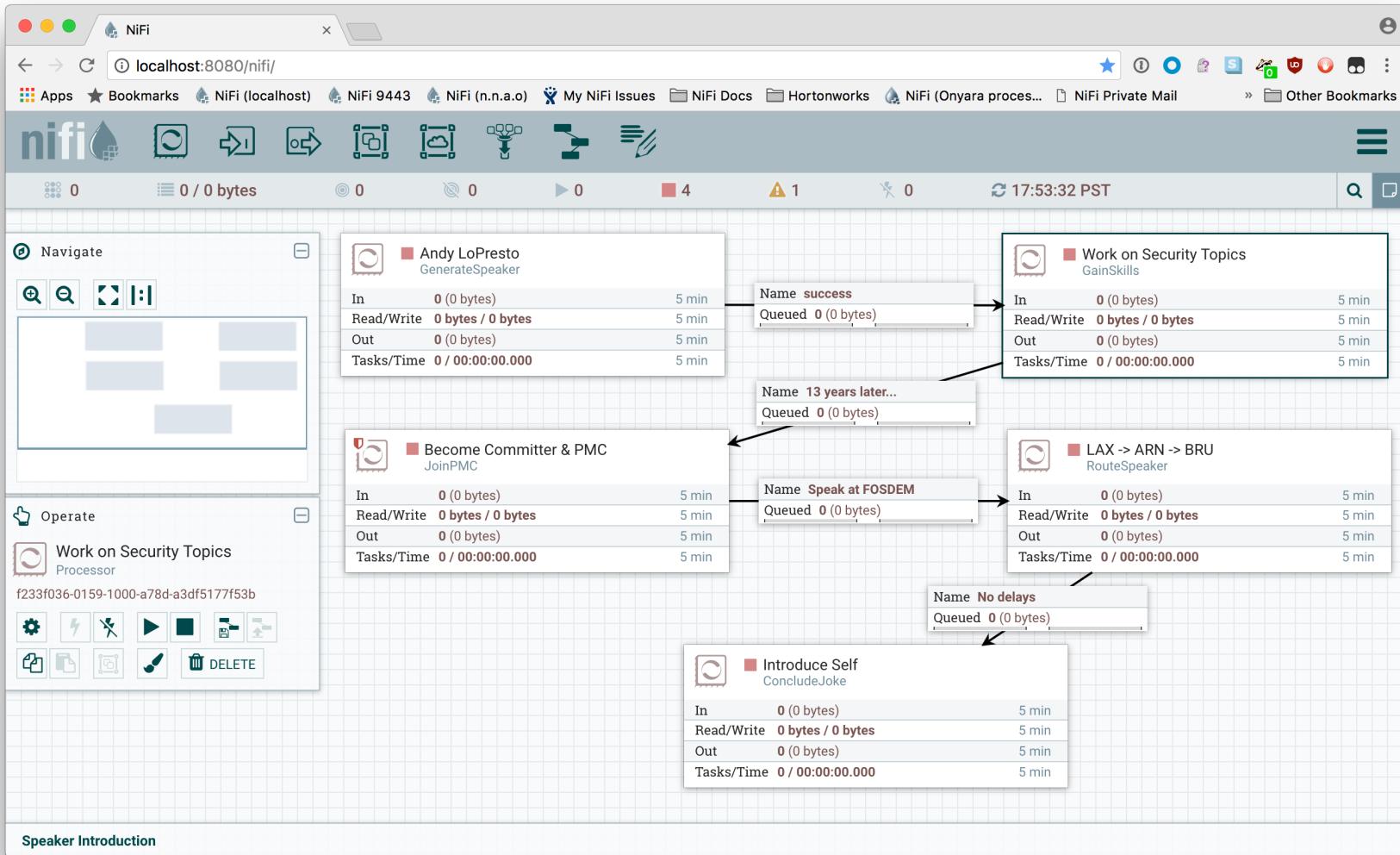
- Agent class
- Runs on minimal hardware
- Interacts with NiFi
- Simple event processing



NiFi Registry

- Standalone application
- Handles asset management
- Flow versioning & deployment
- Extension registry

User Interface



Deeper Ecosystem Integration: 286+ Processors, 61 Controller Services

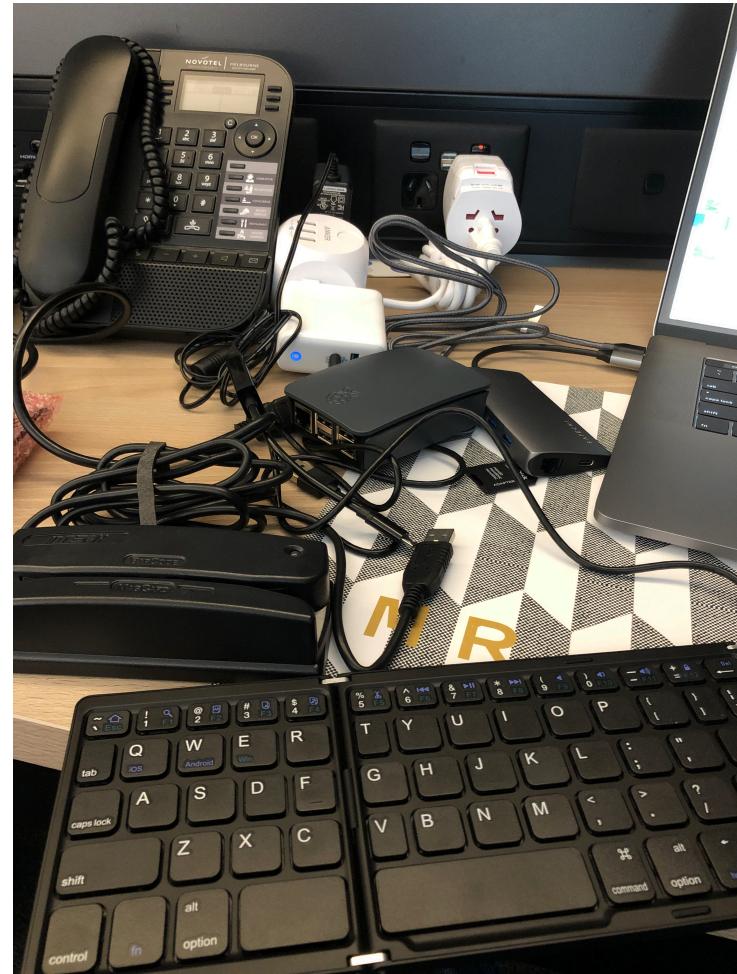


All Apache project logos are trademarks of the ASF and the respective projects.

Extending NiFi functionality

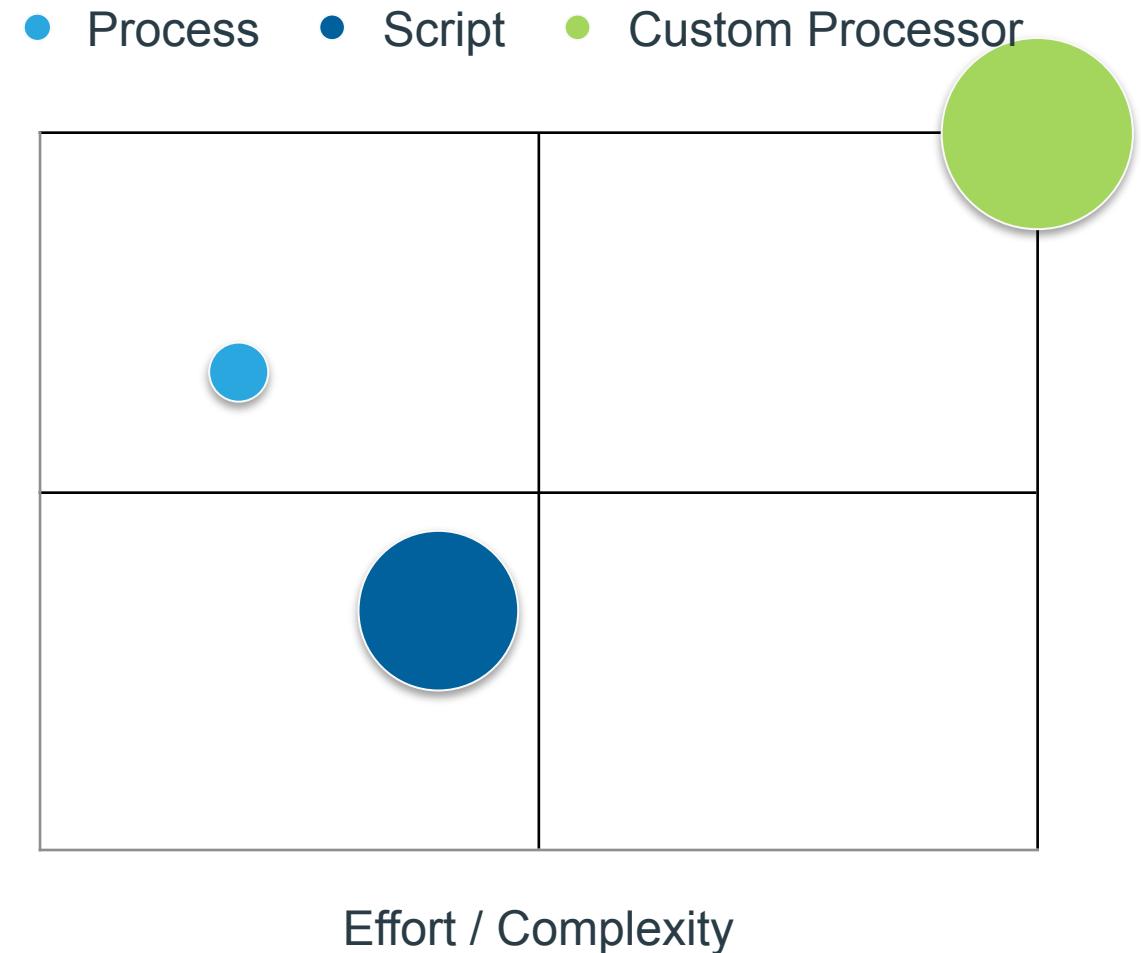
With ~300 processors, what else is left?

- Custom/proprietary protocols
- Read serial interface
 - *USB devices, GPIO on Raspberry Pi*
- Transactional operations
- Experimentation
- Commercial API/SDK
- Legacy code



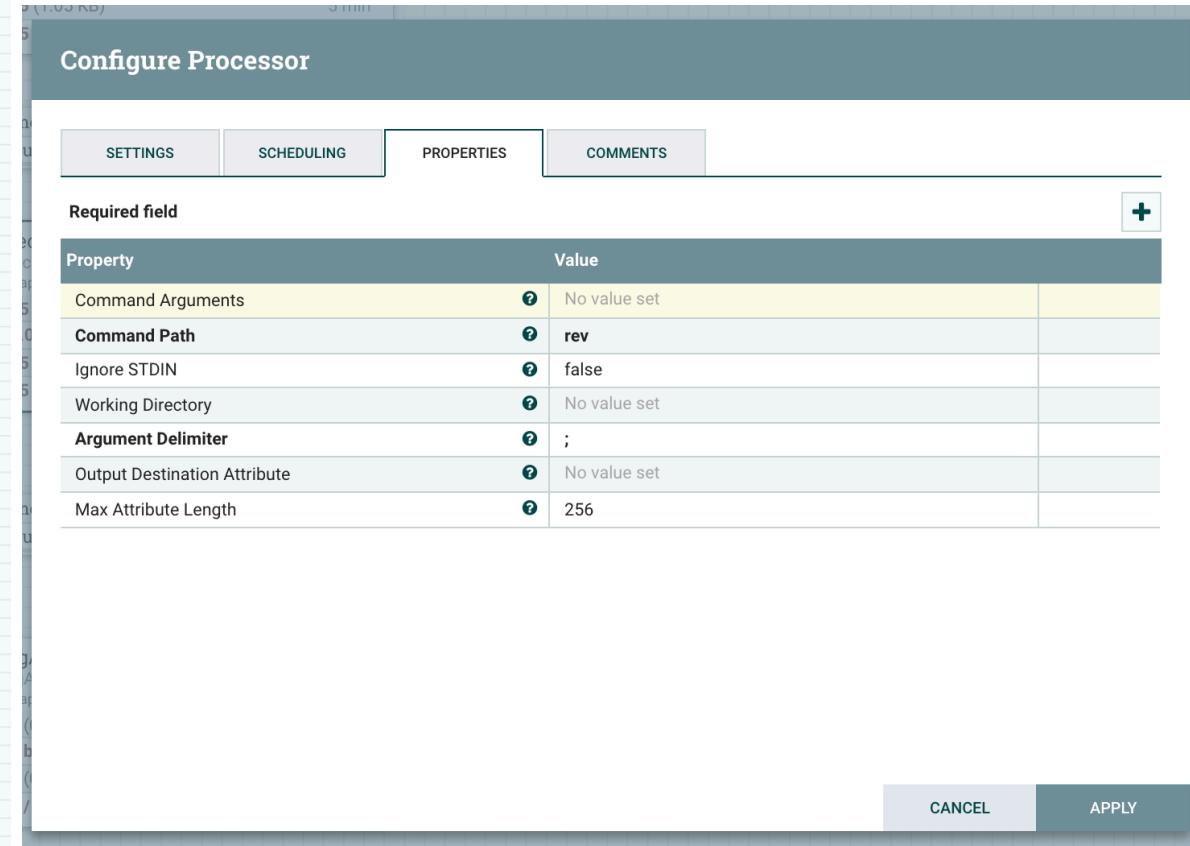
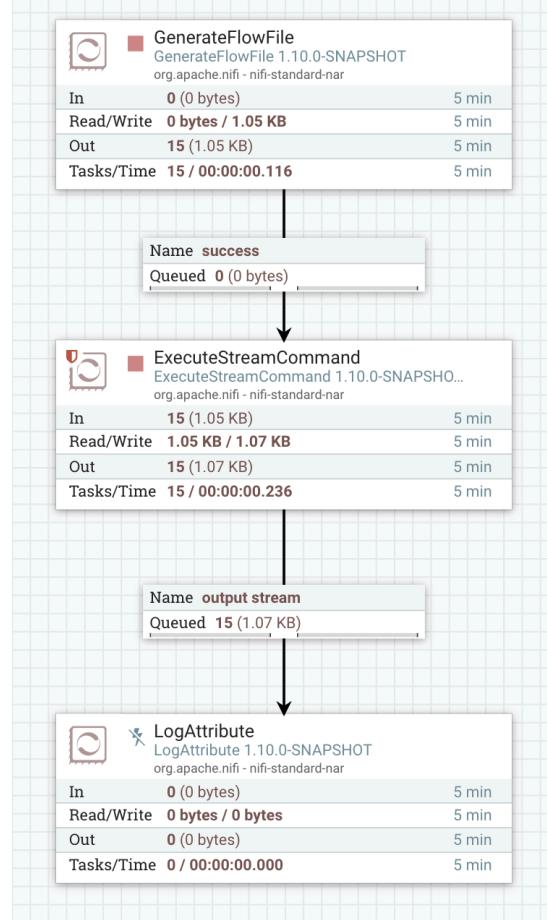
Capabilities of custom development

- ExecuteProcess / ExecuteStreamCommand
- ExecuteScript / InvokeScriptedProcessor
 - ExecuteGroovyScript
- Custom Processor



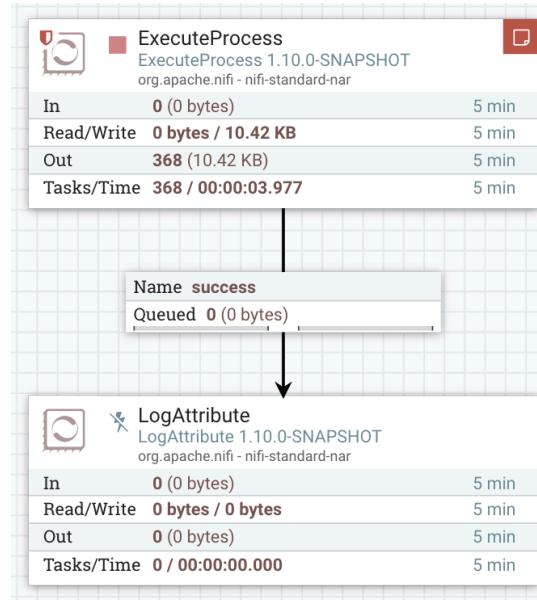
ExecuteStreamCommand

- Pipes flowfile content to STDIN
- Populates outgoing flowfile content from STDOUT
 - Can also direct to named attribute



ExecuteProcess

- Executes arbitrary process and captures output
- *No incoming flowfile*



Configure Processor

SETTINGS SCHEDULING PROPERTIES COMMENTS

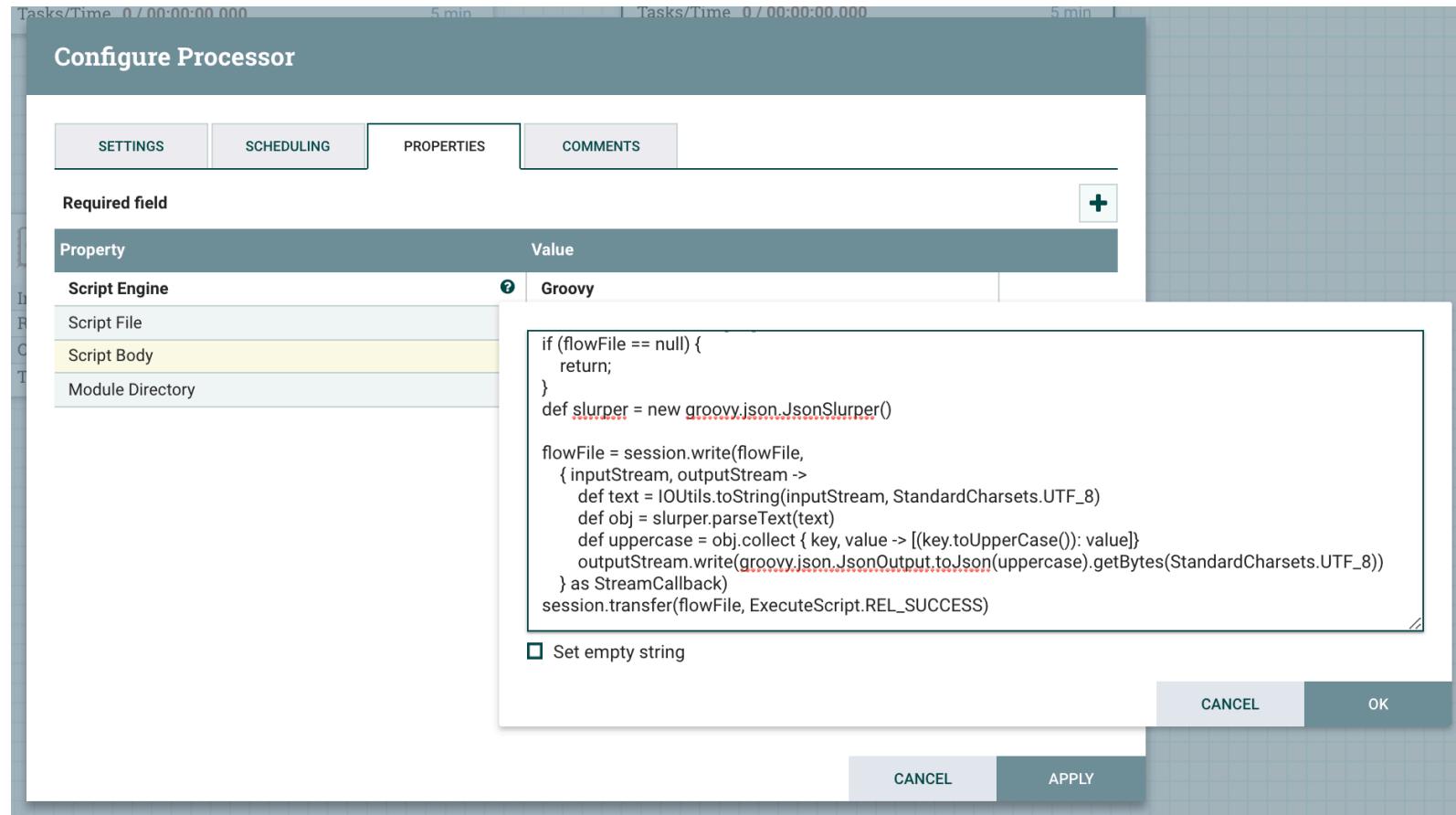
Required field

Property	Value
Command	date
Command Arguments	-u
Batch Duration	No value set
Redirect Error Stream	false
Working Directory	No value set
Argument Delimiter	
source	bash date

CANCEL APPLY

ExecuteScript

- Executes arbitrary code in a JSR-223 compatible language
 - Clojure
 - Groovy
 - Ruby
 - Python (Jython - no C libs)
 - Lua
 - Javascript
- Source is inline or file system



ExecuteGroovyScript

- Community-contributed
- Groovy only
 - Allows for friendlier syntax
 - Allows for Controller Service direct reference

Configure Processor

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field

Property	Value
Script File	
Script Body	
Failure strategy	
Additional classpath	

```
def flowFile = session.get()
if (flowFile == null) {
    return
}

flowFile.write("UTF-8", flowFile.read().getText("UTF-8").reverse())
flowFile.newAttribute = "Some attribute value"
flowFile.reversedAttribute = flowFile.originalAttribute.reverse()
flowFile.transfer(REL_SUCCESS)
```

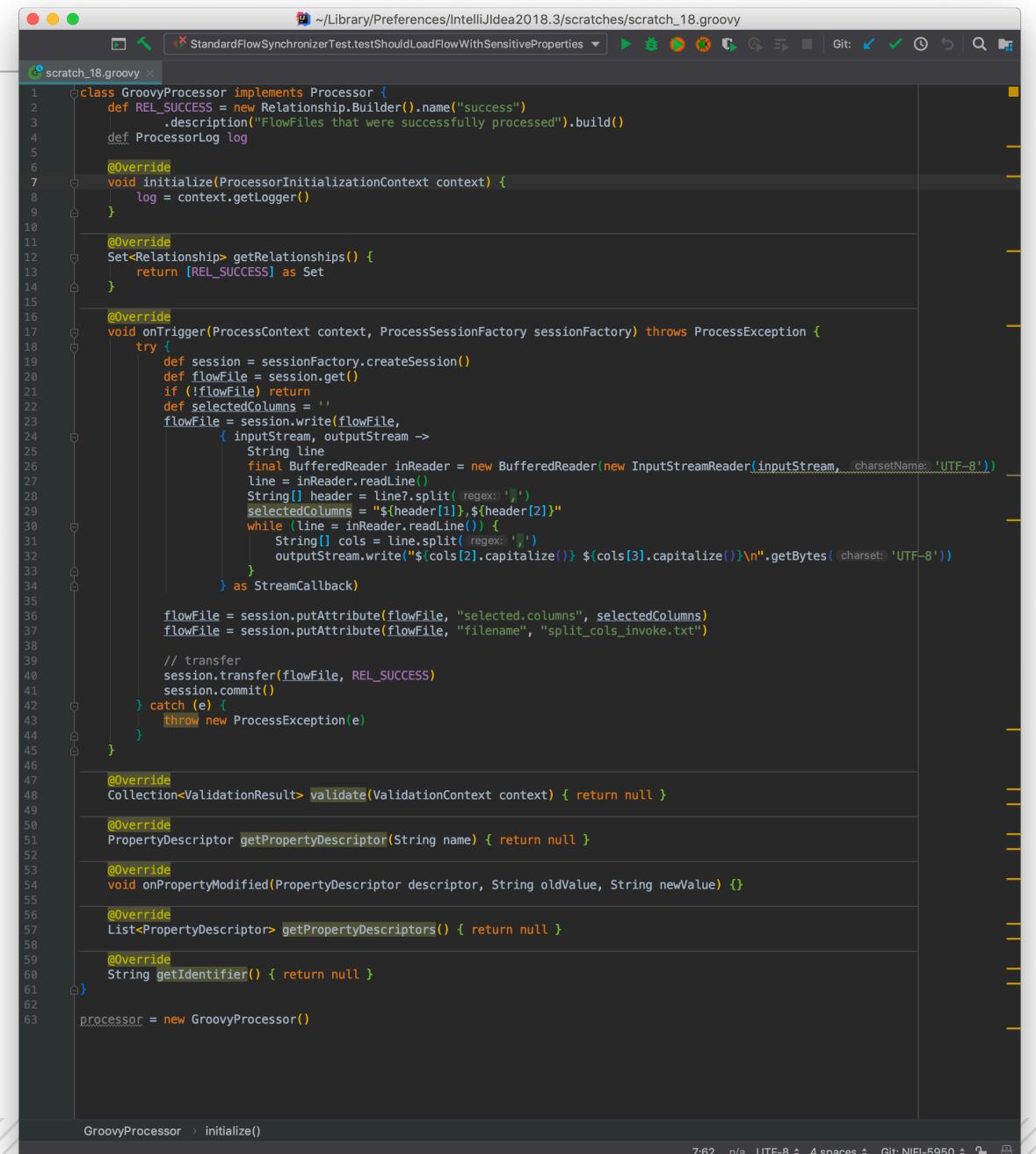
Set empty string

CANCEL OK

CANCEL APPLY

InvokeScriptedProcessor

- “*ExecuteScript = onTrigger()*”
- Allows for additional methods
- Not compiled for each flowfile
- Supports custom relationships
- Improves performance



The screenshot shows a code editor window titled "scratch_18.groovy". The code is a Groovy class named "GroovyProcessor" that implements the "Processor" interface. The class contains several overridden methods: "initialize", "getRelationships", "onTrigger", "validate", "getPropertyDescriptor", "onPropertyModified", "getPropertyDescriptor", and "getIdentifier". The "onTrigger" method is annotated with "@Override" and contains logic to read a header from a flowfile, split it into columns, and then write the modified data back to the flowfile. The "validate" method returns null. The "getPropertyDescriptor" and "onPropertyModified" methods also return null. The "getPropertyDescriptor" and "getPropertyDescriptor" methods both return a list containing null. The "getIdentifier" method returns null. The "processor" variable is initialized to a new instance of "GroovyProcessor".

```
class GroovyProcessor implements Processor {
    def REL_SUCCESS = new Relationship.Builder().name("success")
        .description("FlowFiles that were successfully processed").build()
    def ProcessorLog log

    @Override
    void initialize(ProcessorInitializationContext context) {
        log = context.getLogger()
    }

    @Override
    Set<Relationship> getRelationships() {
        return [REL_SUCCESS] as Set
    }

    @Override
    void onTrigger(ProcessContext context, ProcessSessionFactory sessionFactory) throws ProcessException {
        try {
            def session = sessionFactory.createSession()
            def flowFile = session.get()
            if (!flowFile) return
            if (selectedColumns == '') {
                flowFile = session.write(flowFile,
                    { inputStream, outputStream -->
                        String line
                        final BufferedReader inReader = new BufferedReader(new InputStreamReader(inputStream, charsetName: 'UTF-8'))
                        line = inReader.readLine()
                        String[] header = line.split( regex: ',' )
                        selectedColumns = "${header[1]},${header[2]}"
                        while (line = inReader.readLine()) {
                            String[] cols = line.split( regex: ',' )
                            outputStream.write("${cols[2].capitalize()} ${cols[3].capitalize()}\n".getBytes( charset: 'UTF-8' ))
                        }
                    } as StreamCallback)
            }
            flowFile = session.putAttribute(flowFile, "selected.columns", selectedColumns)
            flowFile = session.putAttribute(flowFile, "filename", "split_cols_invoke.txt")

            // transfer
            session.transfer(flowFile, REL_SUCCESS)
            session.commit()
        } catch (e) {
            throw new ProcessException(e)
        }
    }

    @Override
    Collection<ValidationResult> validate(ValidationContext context) { return null }

    @Override
   PropertyDescriptor getPropertyDescriptor(String name) { return null }

    @Override
    void onPropertyModified(PropertyDescriptor descriptor, String oldValue, String newValue) {}

    @Override
    List<PropertyDescriptor> getPropertyDescriptors() { return null }

    @Override
    String getIdentifier() { return null }

    processor = new GroovyProcessor()
}
```

Custom processor development

Custom Processor

- High performance
- Reusability & convenience
- Versioning

Add Processor

Source: com.andylolopresto

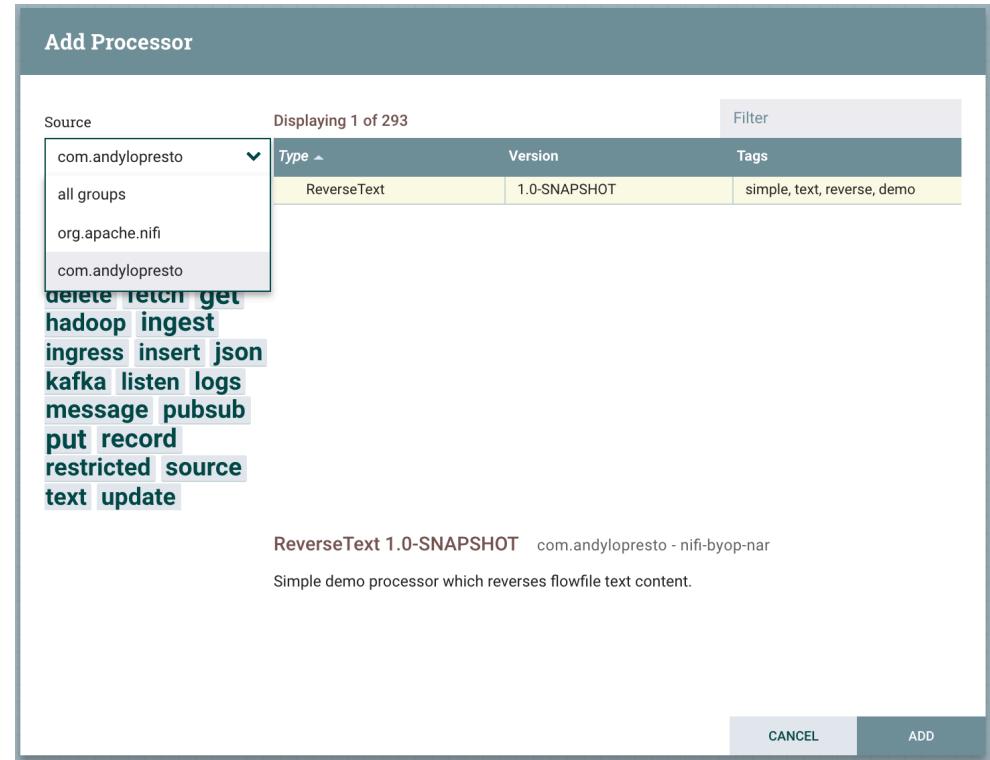
Type: ReverseText

Version: 1.0-SNAPSHOT

Tags: simple, text, reverse, demo

ReverseText 1.0-SNAPSHOT com.andylolopresto - nifi-byop-nar
Simple demo processor which reverses flowfile text content.

CANCEL ADD



ReverseText

ReverseText 1.0-SNAPSHOT

com.andylolopresto - nifi-byop-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min



Using the Maven Archetype

- Prompts for metadata input
 - groupId
 - artifactId
 - version
 - artifactBaseName
 - package
- Generates a Maven project structure

```
1. ...1.10.0-SNAPSHOT-bin/nifi-1.10.0-SNAPSHOT (bash)
...1.10.0-SNAPSHOT-bin/nifi-1.10.0-SNAPSHOT (bash) 29s @ 20:51:01 $ mvn archetype:generate -DarchetypeGroupId=org.apache.nifi -DarchetypeArtifactId=nifi-processor-bundle-archetype
[INFO] archetypeVersion=1.9.0 -DnifiVersion=1.9.0
[INFO] Scanning for projects...
[INFO] < org.apache.maven:standalone-pom >
[INFO] Building Maven Stub Project (No POM) 1
[INFO] [ pom ]
[INFO]
[INFO] >>> maven-archetype-plugin:3.0.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.0.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.0.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[INFO] Archetype repository not defined. Using the one from [org.apache.nifi:nifi-processor-bundle-archetype:1.0.0.2.0.0.0-265 -> http://nexus-private.hortonworks.com/nexus/content/groups/public] found in catalog remote
Define value for property 'groupId': com.andylopresto
Define value for property 'artifactId': nifi-byop-dws
Define value for property 'version' 1.0-SNAPSHOT: :
Define value for property 'artifactBaseName': byop
Define value for property 'package' com.andylopresto.processors.byop: :
[INFO] Using property: nifiVersion = 1.9.0
Confirm properties configuration:
groupId: com.andylopresto
artifactId: nifi-byop-dws
version: 1.0-SNAPSHOT
artifactBaseName: byop
package: com.andylopresto.processors.byop
nifiVersion: 1.9.0
Y: : Y
[INFO]
[INFO] Using following parameters for creating project from Archetype: nifi-processor-bundle-archetype:1.9.0
[INFO]
[INFO] Parameter: groupId, Value: com.andylopresto
[INFO] Parameter: artifactId, Value: nifi-byop-dws
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.andylopresto.processors.byop
[INFO] Parameter: packageInPathFormat, Value: com.andylopresto/processors/byop
[INFO] Parameter: package, Value: com.andylopresto.processors.byop
[INFO] Parameter: artifactBaseName, Value: byop
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: groupId, Value: com.andylopresto
[INFO] Parameter: artifactId, Value: nifi-byop-dws
[INFO] Parameter: nifiVersion, Value: 1.9.0
[INFO] Project created from Archetype in dir: /Users/alopresto/Workspace/nifi/nifi-assembly/target/nifi-1.10.0-SNAPSHOT-bin/nifi-1.10.0-SNAPSHOT/nifi-byop-dws
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 53.326 s
[INFO] Finished at: 2019-03-12T20:55:00-07:00
[INFO]
```

Results of the Maven Archetype

- **MyProcessor.java**
 - Source code for processor
- **org.apache.nifi.processor.Processor**
 - Manifest file listing implementations
- **MyProcessorTest.java**
 - JUnit test
- **pom.xml**
 - Maven Project Object Model file

```
...lopreno/Workspace/scratch/nifi-byop-dws (master) 😊
└─ 0s @ 20:56:18 $ tl
.
├── [ 96] nifi-byop-nar/
│   └── [1.7K] pom.xml
├── [ 128] nifi-byop-processors/
│   └── [2.1K] pom.xml
└── [ 128] src/
    └── [ 128] main/
        ├── [ 96] java/
        │   └── [ 96] com/
        │       └── [ 96] andylopreno/
        │           └── [ 96] processors/
        │               └── [ 96] byop/
        │                   └── [3.8K] MyProcessor.java
        └── [ 96] resources/
            └── [ 96] META-INF/
                └── [ 96] services/
                    └── [ 825] org.apache.nifi.processor.Processor
    └── [ 96] test/
        └── [ 96] java/
            └── [ 96] com/
                └── [ 96] andylopreno/
                    └── [ 96] processors/
                        └── [ 96] byop/
                            └── [1.2K] MyProcessorTest.java
└── [1.5K] pom.xml
18 directories, 6 files
...lopreno/Workspace/scratch/nifi-byop-dws (master) 😊
└─ 0s @ 20:59:59 $
```

Metadata

- **@Tags**
 - Helpful for discovery
- **@CapabilityDescription**
 - Shown in documentation
 - Annotations for behavior
- **@SeeAlso**
 - References similar/complementary processors
- **@Reads/WritesAttributes**
 - Listed in documentation

```
43  @Tags({"example"})
44  @CapabilityDescription("Provide a description")
45  @SeeAlso({})
46  @ReadsAttributes({@ReadsAttribute(attribute="", description(""))})
47  @WritesAttributes({@WritesAttribute(attribute="", description(""))})
48  public class MyProcessor extends AbstractProcessor {
```

Selected other annotations

- **@EventDriven**
 - Indicates not based on timer
- **@SideEffectFree**
 - Indicates safe to run w/o side effects
- **@SupportsBatching**
 - Indicates multiple flowfiles processed together
- **@InputRequirement**
 - Can't be the first processor in a flow
- **@SystemResourceConsideration**
 - Warns users of implications
- **@Experimental**
 - Self-explanatory
- **@Restricted**
 - Controls access

```
67     @EventDriven
68     @SideEffectFree
69     @SupportsBatching
70     @InputRequirement(Requirement.INPUT_REQUIRED)
71     @Tags({"encryption", "decryption", "password", "JCE", "OpenPGP", "PGP", "GPG"})
72     @CapabilityDescription("Encrypts or Decrypts a FlowFile using either symmetric
73     generated salt, or asymmetric encryption using a public and secret key.")
74     @SystemResourceConsideration(resource = SystemResource.CPU)
    public class EncryptContent extends AbstractProcessor {
```

Implementing the code

- Add property descriptors and relationships
- Implement `onTrigger()` method
 - This is where the logic is done
 - Reads properties & attributes
 - Configures external services
 - Executes data transformation
 - Transfers flowfile
- Add custom `StreamCallback` class
 - Handles “behavior” of reversing String
- “32” lines of code

```
1  /...
17 package com.andylopresto.processors.byop;
18
19 import ...
41
42 @Tags({"simple", "text", "demo", "reverse"})
43 @CapabilityDescription("Simple demo processor which reverses flowfile text content. ")
44 public class ReverseText extends AbstractProcessor {
45     public static final Relationship REL_SUCCESS = new Relationship.Builder()
46         .name("REL_SUCCESS")
47         .description("Success relationship")
48         .build();
49
50     private List<PropertyDescriptor> descriptors;
51
52     private Set<Relationship> relationships;
53
54     @Override
55     protected void init(final ProcessorInitializationContext context) {
56         this.descriptors = Collections.emptyList();
57
58         final Set<Relationship> relationships = new HashSet<~>();
59         relationships.add(REL_SUCCESS);
60         this.relationships = Collections.unmodifiableSet(relationships);
61     }
62
63     @Override
64     public Set<Relationship> getRelationships() { return this.relationships; }
65
66     @Override
67     public final List<PropertyDescriptor> getSupportedPropertyDescriptors() { return descriptors; }
68
69     @OnScheduled
70     public void onScheduled(final ProcessContext context) {}
71
72     @Override
73     public void onTrigger(final ProcessContext context, final ProcessSession session) throws ProcessException {
74         FlowFile flowFile = session.get();
75         if ( flowFile == null ) {
76             return;
77
78             // TODO: This should check the mime/type if available, and the content encoding to attempt to detect non-text content
79
80             // TODO: This should be done in a streaming manner
81
82             session.write(flowFile, new ReverseTextCallback());
83             session.putAttribute(flowFile, s: "reversed", s1: "true");
84             session.transfer(flowFile, REL_SUCCESS);
85         }
86     }
87
88     class ReverseTextCallback implements StreamCallback {
89
90         @Override
91         public void process(InputStream inputStream, OutputStream outputStream) throws IOException {
92             // This example only reads the first line
93             String originalContent = new BufferedReader(new InputStreamReader(inputStream)).readLine();
94             outputStream.write(new StringBuilder(originalContent).reverse().toString().getBytes(StandardCharsets.UTF_8));
95         }
96     }
97
98 }
99
100
101
102
103
104 }
```

Testing

Write the tests (should be first)

- Test-driven development (TDD)
- Helpful **TestRunner** class
 - Handles enqueueing arbitrary flowfile content & attributes
 - Allows intelligent assertions

```
#!/...
package com.andylopresto.processors.byop;

import ...

public class ReverseTextTest {
    private static final Logger logger = LoggerFactory.getLogger(ReverseTextTest.class);
    private static final String MESSAGE = "This is a plaintext message. ";
    private static final String REVERSED_MESSAGE = new StringBuilder(MESSAGE).reverse().toString();

    private TestRunner testRunner;

    @Before
    public void init() { testRunner = TestRunners.newTestRunner(ReverseText.class); }

    @Test
    public void testProcessor() {
        // Arrange
        testRunner.enqueue(MESSAGE);

        // Act
        testRunner.run();
        logger.info("Ran once with " + MESSAGE);

        // Assert
        testRunner.assertAllFlowFilesTransferred(ReverseText.REL_SUCCESS, 1);
        MockFlowFile flowFile = testRunner.getFlowFilesForRelationship(ReverseText.REL_SUCCESS).get(0);
        flowFile.assertContentEquals(REVERSED_MESSAGE);
        flowFile.assertAttributeEquals("reversed", "true");
    }
}

Tests passed: 1 of 1 test - 124 ms
/Library/Java/JavaVirtualMachines/jdk1.8.0_192.jdk/Contents/Home/bin/java ...
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/Users/alopresto/.m2/repository/ch/qos/logback/logback-classic/1.2.3/logback-classic-1.2.3.jar!$logback-classic.jar]
SLF4J: Found binding in [jar:file:/Users/alopresto/.m2/repository/org/slf4j/slf4j-simple/1.7.25/slf4j-simple-1.7.25.jar!$slf4j-simple.jar]
SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.
SLF4J: Actual binding is of type [ch.qos.logback.classic.util.ContextSelectorStaticBinder]
21:46:17.497 [main] INFO com.andylopresto.processors.byop.ReverseTextTest - Ran once with This is a plaintext message.

Process finished with exit code 0
```

We'll come back to this...

Deploying

Build the Maven project

- `mvn clean install`
- Builds the project
 - Compiles source code
 - Compiles tests
 - Runs tests
 - Outputs target file(s)

```
[INFO] -----
[INFO] Reactor Summary for nifi-byop-dws 1.0-SNAPSHOT:
[INFO]
[INFO] nifi-byop-dws ..... SUCCESS [ 2.263 s]
[INFO] nifi-byop-processors .. SUCCESS [ 2.957 s]
[INFO] nifi-byop-nar ..... SUCCESS [ 0.242 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.125 s
[INFO] Finished at: 2019-03-12T22:01:00-07:00
[INFO] -----
```

```
>.../lopresto/Workspace/scratch/nifi-byop-dws (master) 😊
└─ 9s @ 22:01:00 $ ll nifi-byop-nar/target/
total 376
drwxr-xr-x 12 alopresto staff 384B Mar 12 22:01 .
drwxr-xr-x  4 alopresto staff 128B Mar 12 22:01 ..
-rw-r--r--  1 alopresto staff  30B Mar 12 22:01 .plxarc
-rw-r--r--  1 alopresto staff 2.8K Mar 12 22:01 checkstyle-checker.xml
-rw-r--r--  1 alopresto staff  81B Mar 12 22:01 checkstyle-result.xml
-rw-r--r--  1 alopresto staff 2.8K Mar 12 22:01 checkstyle-rules.xml
drwxr-xr-x  3 alopresto staff  96B Mar 12 22:01 classes/
drwxr-xr-x  3 alopresto staff  96B Mar 12 22:01 maven-archiver/
drwxr-xr-x  3 alopresto staff  96B Mar 12 22:01 maven-shared-archive-resources/
-rw-r--r--  1 alopresto staff 164K Mar 12 22:01 nifi-byop-nar-1.0-SNAPSHOT.nar
-rw-r--r--  1 alopresto staff 729B Mar 12 22:01 rat.txt
drwxr-xr-x  3 alopresto staff  96B Mar 12 22:01 test-classes/
```

Copy the NAR

- Copy into NiFi's library
 - **\$NIFI_HOME/lib**
 - NiFi will load extension automatically during startup

```
cked
    org.apache.nifi.processors.hadoop.ListHDFS
        org.apache.nifi:nifi-hadoop-nar:1.10.0-SNAPSHOT || ./work/nar/extensions/nifi-hadoop-nar-1.10.0-SNAPSHOT.nar-unpacked
    org.apache.nifi.processors.kafka.GetKafka
        org.apache.nifi:nifi-kafka-0-8-nar:1.10.0-SNAPSHOT || ./work/nar/extensions/nifi-kafka-0-8-nar-1.10.0-SNAPSHOT.nar-unpacked
    com.andyllopreno.processors.byop.ReverseText
        com.andyllopreno:nifi-byop-nar:1.0-SNAPSHOT || ./work/nar/extensions/nifi-byop-nar-1.0-SNAPSHOT.nar-unpacked
    org.apache.nifi.processors.gcp.storage.DeleteGCSObject
        org.apache.nifi:nifi-gcp-nar:1.10.0-SNAPSHOT || ./work/nar/extensions/nifi-gcp-nar-1.10.0-SNAPSHOT.nar-unpacked
    org.apache.nifi.processors.standard.UpdateRecord
        org.apache.nifi:nifi-standard-nar:1.10.0-SNAPSHOT || ./work/nar/extensions/nifi-standard-nar-1.10.0-SNAPSHOT.nar-unpacked
    org.apache.nifi.processors.elasticsearch.ScrollElasticsearchHttp
        org.apache.nifi:nifi-elasticsearch-nar:1.10.0-SNAPSHOT || ./work/nar/extensions/nifi-elasticsearch-nar-1.10.0-SNAPSHOT.nar-unpacked
```

Add the processor

- Can filter by source group
- Can search by tags or processor name

Add Processor

Source	Type	Version	Tags
com.andylopresto	ReverseText	1.0-SNAPSHOT	simple, text, reverse, demo

Displaying 1 of 293

Filter

Source dropdown menu:

- com.andylopresto
- all groups
- org.apache.nifi
- com.andylopresto

Processor tags:

- delete
- fetch
- get
- hadoop
- ingest
- ingress
- insert
- json
- kafka
- listen
- logs
- message
- pubsub
- put
- record
- restricted
- source
- text
- update

Processor details:

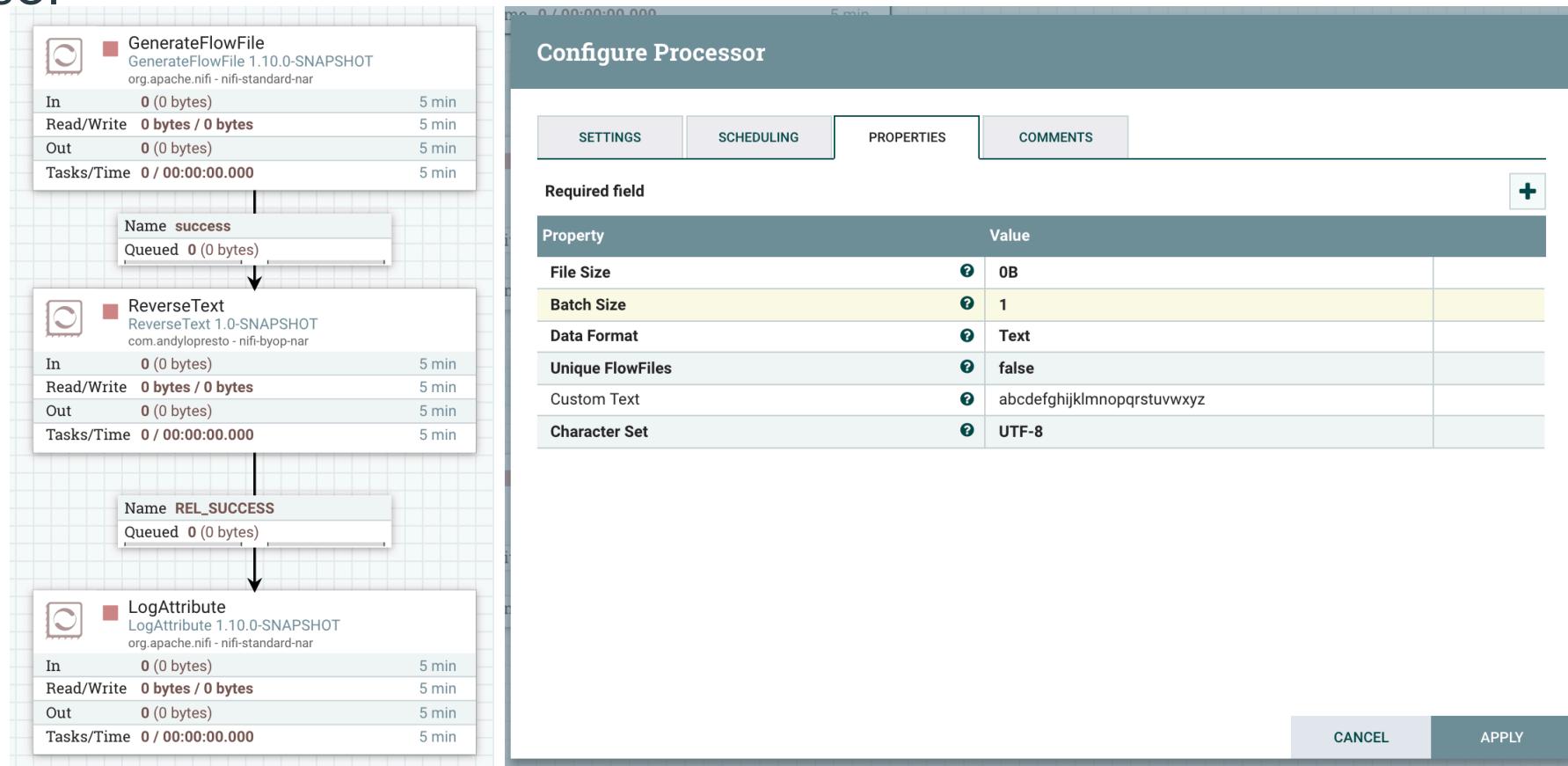
ReverseText 1.0-SNAPSHOT com.andylopresto - nifi-byop-nar

Simple demo processor which reverses flowfile text content.

CANCEL ADD

Configure the processor

- Feed data with GenerateFlowFile



Smoke test

- List queue on the success relationship
- View attribute
 - “reversed” - true
- View content

DETAILS ATTRIBUTES

Attribute Values

filename	3a779fb6-3965-4c56-b5cb-e75aeb78f7c0
path	./
reversed	true
uuid	3a779fb6-3965-4c56-b5cb-e75aeb78f7c0

View as: original ▾

1 zyxwvutsrqponmlkjihgfedcba

NAR loading

- NiFi now hot-loads custom NARs in `$NIFI_HOME/extensions`
- Not good for SDLC
 - Doesn't reload matching NAR
 - *Could use build script to append timestamp to filename and version*

```
...1.10.0-SNAPSHOT-bin/nifi-1.10.0-SNAPSHOT (byop-demo) 😊
└─ 0s @ 23:08:56 $ ll
total 424
drwxr-xr-x  18 alopreno staff  576B Mar 12 22:07 .
drwxr-xr-x   3 alopreno staff   96B Mar 12 20:41 ..
-rw-r--r--   1 alopreno staff 119K Aug 21 2018 LICENSE
-rw-r--r--   1 alopreno staff  81K Feb  6 16:12 NOTICE
-rw-r--r--   1 alopreno staff  4.4K Nov 19 14:48 README
drwxr-xr-x   8 alopreno staff 256B Mar 12 20:41 bin/
drwxr-xr-x  12 alopreno staff 384B Mar 12 22:43 conf/
drwxr-xr-x 1026 alopreno staff  32K Mar 12 22:07 content_repository/
drwxr-xr-x   6 alopreno staff 192B Mar 12 22:07 database_repository/
drwxr-xr-x   3 alopreno staff  96B Mar 12 20:41 docs/
drwxrwxr-x   2 alopreno staff  64B Mar 12 20:41 extensions/
drwxr-xr-x   5 alopreno staff 160B Mar 12 22:15 flowfile_repository/
drwxr-xr-x  120 alopreno staff  3.8K Mar 12 22:06 lib/
drwxr-xr-x   6 alopreno staff 192B Mar 12 23:01 logs/
drwxr-xr-x   5 alopreno staff 160B Mar 12 22:14 provenance_repository/
drwxr-xr-x   4 alopreno staff 128B Mar 12 22:06 run/
drwxr-xr-x   3 alopreno staff  96B Mar 12 22:07 state/
drwxr-xr-x   5 alopreno staff 160B Mar 12 22:07 work/
```

Testing (continued)

Testing approaches

- Unit testing
- Integration testing
- In-NiFi testing

Unit testing

- Regression testing
 - Supports refactoring
 - Supports new feature development
- Extract behavior to service class
 - Tests for logic and for “processor” behavior
 - Combinations of properties & attributes/content
- Use the mock classes
 - **MockFlowFile** - allows for internal assertions
 - **MockProcessSession** - allows for flowfile operation (create, transfer, clone, etc.)
 - **MockProcessContext** - allows for property and controller service interaction

Groovy unit testing

- Easy mocking
 - Map coercion
 - Metaclass overrides
- Sparser syntax; less boilerplate
- Spock for BDD

```
Security.addProvider(new BouncyCastleProvider())

logger.metaClass.methodMissing = { String name, args ->
    logger.info("[${name?.toUpperCase()}] ${args as List}.join(" ")}}

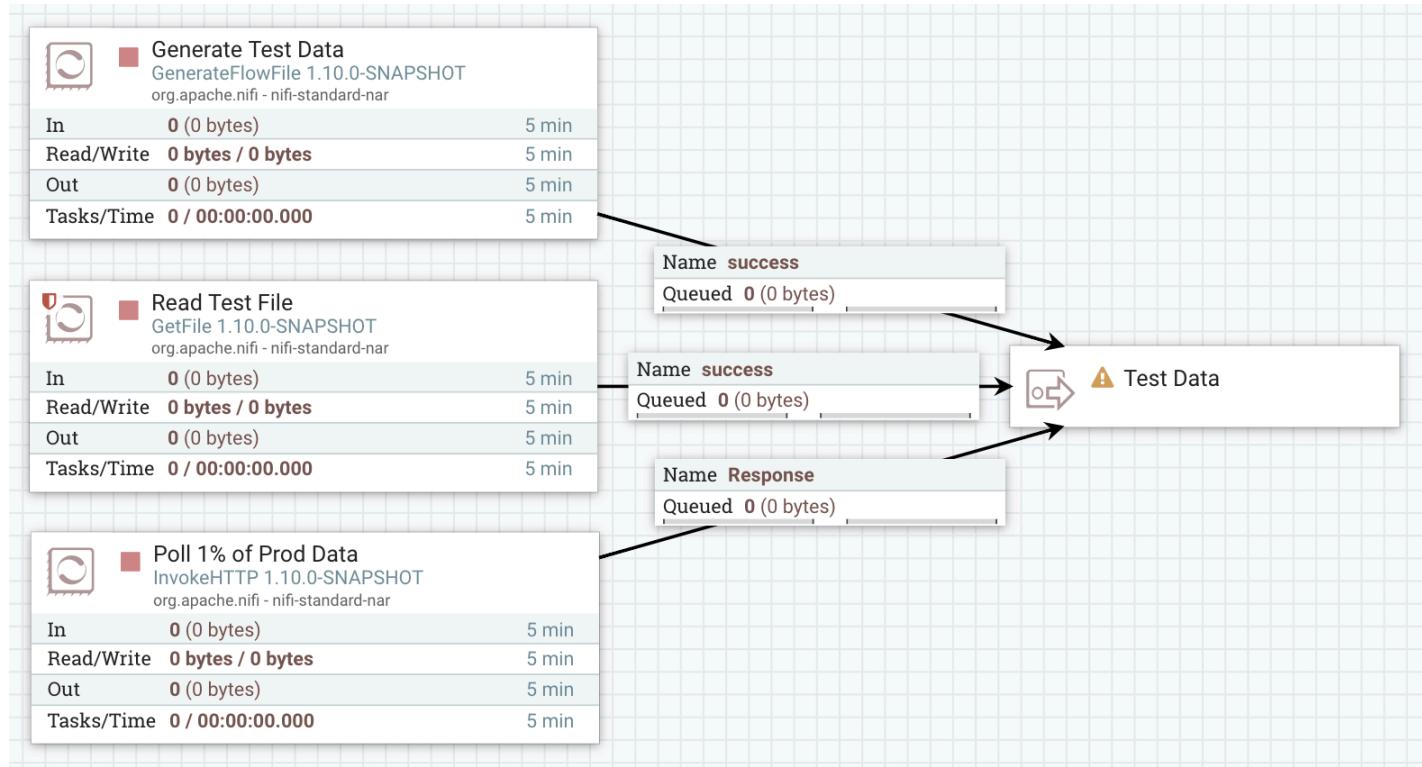
mockKeyProvider = [
    getKey   : { String keyId ->
        logger.mock("Requesting key ID: ${keyId}")
        new SecretKeySpec(Hex.decode(KEY_HEX), "AES")
    },
    keyExists: { String keyId ->
        logger.mock("Checking existence of ${keyId}")
        true
    }
] as KeyProvider
```

Integration testing

- Provide external services for testing
 - Docker containers are popular
 - Easy to spin up & delete
- Example
 - `MongoDBLookupServiceIT`

Testing inside NiFi

- Create reusable process group which handles test fixtures
 - Use **GenerateFlowFile**
 - Call external APIs
 - Read from test data on file system
 - Read from test database
 - Connect output port to “flow under test”



Other capabilities

Custom User Interface

- Sometimes generic property descriptors don't make sense
 - Niels Basjes' logparser project with 100+ boolean flags that are auto-detected

<https://github.com/niebsbasjes/logparser>

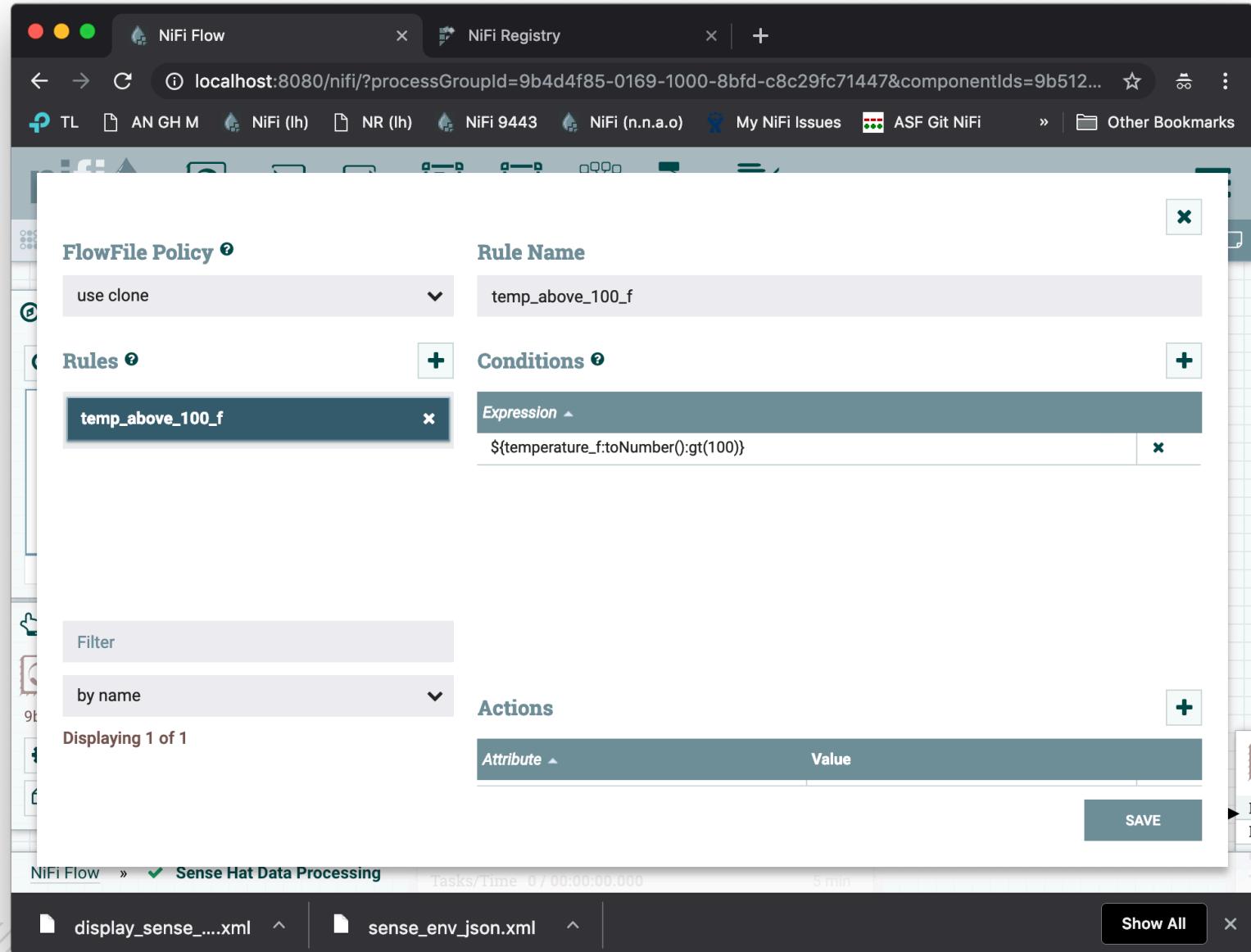
The dialog shows the 'Processor Details' screen with the 'PROPERTIES' tab selected. It displays a table of properties with their values. One property, 'Store State', is highlighted as a 'Required field'. The table has two columns: 'Property' and 'Value'.

Property	Value
Delete Attributes Expression	?
Store State	?
Stateful Variables Initial Value	?
Cache Value Lookup Cache Size	?
temperature_f	?

At the bottom left is an 'ADVANCED' button, and at the bottom right is an 'OK' button.

Custom User Interface

- Allows for custom UX
- Validation, feedback, wizards, etc.



Custom User Interface

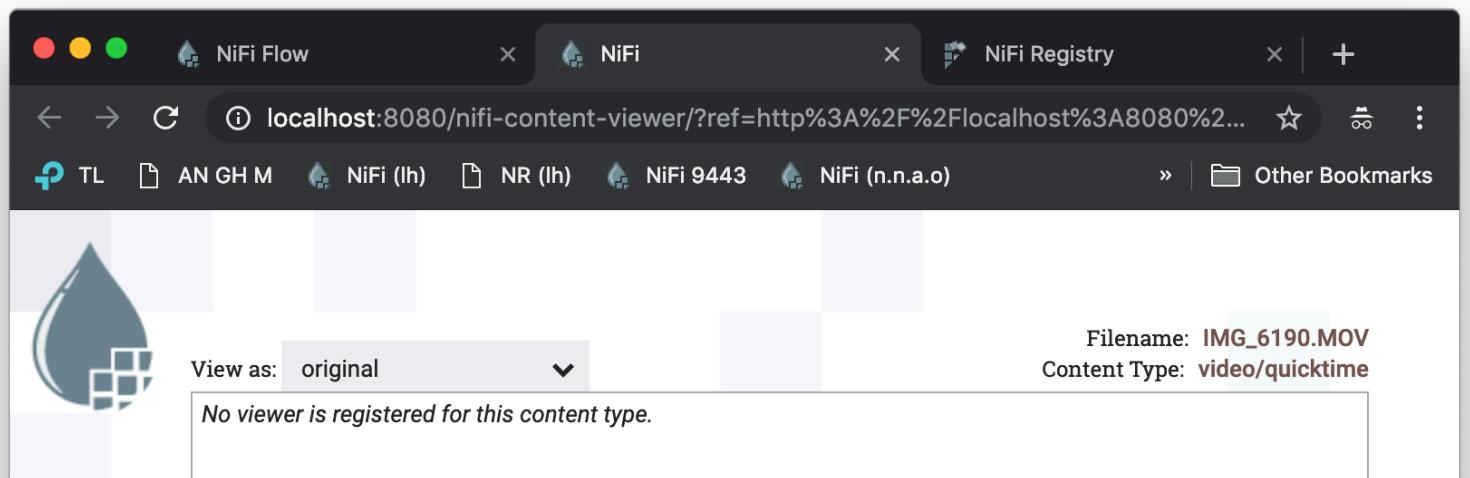
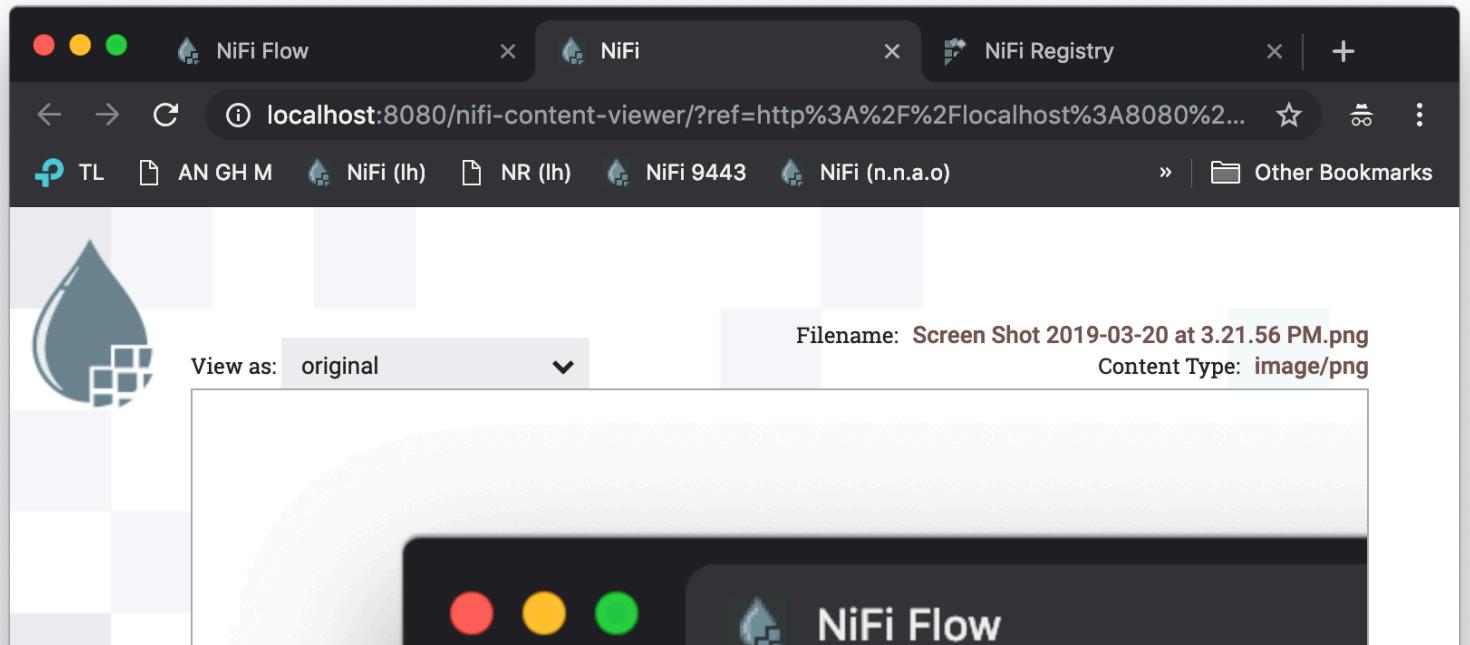
- Code in `nifi-processor-custom-ui`
 - Java classes for objects & logic
 - CSS/JS for presentation

```
1 ...ttribute-bundle/nifi-update-attribute-ui (bash)
...ttribute-bundle/nifi-update-attribute-ui (byop-demo) 😊
3s @ 15:07:36 $ tl
.
└── [6.7K] nifi-update-attribute-ui.iml
   ├── [5.8K] pom.xml
   └── [ 96] src/
       ├── [ 160] main/
       │   ├── [ 96] java/
       │   │   └── [ 96] org/
       │   │       └── [ 96] apache/
       │   │           └── [ 96] nifi/
       │   │               └── [ 96] update/
       │   │                   ├── [ 192] attributes/
       │   │                       ├── [ 5.0K] UpdateAttributeModelFactory.java
       │   │                       └── [ 96] api/
       │   │                           └── [ 28K] RuleResource.java
       │   │                   ├── [ 192] dto/
       │   │                       ├── [ 2.1K] ActionDTO.java
       │   │                       ├── [ 1.9K] ConditionDTO.java
       │   │                       ├── [ 2.3K] DtoFactory.java
       │   │                       └── [ 1.7K] RuleDTO.java
       │   │                   ├── [ 224] entity/
       │   │                       ├── [ 1.7K] ActionEntity.java
       │   │                       ├── [ 1.8K] ConditionEntity.java
       │   │                       ├── [ 2.3K] EvaluationContextEntity.java
       │   │                       ├── [ 2.0K] RuleEntity.java
       │   │                       └── [ 1.8K] RulesEntity.java
       │   └── [ 96] resources/
       │       ├── [ 128] META-INF/
       │       │   ├── [ 13K] LICENSE
       │       │   └── [ 3.2K] NOTICE
       └── [ 224] webapp/
           ├── [ 96] META-INF/
           │   ├── [ 899] nifi-processor-configuration*
           │   └── [ 128] WEB-INF/
           │       ├── [ 96] jsp/
           │       │   └── [ 13K] worksheet.jsp
           │       └── [ 2.1K] web.xml
           └── [ 96] css/
               └── [ 6.2K] main.css
           └── [ 192] images/
               ├── [ 139] bgInputText.png*
               ├── [ 573] buttonNew.png
               ├── [ 555] iconDelete.png
               ├── [ 414] iconInfo.png
               └── [ 96] js/
                   └── [ 73K] application.js

20 directories, 24 files
...ttribute-bundle/nifi-update-attribute-ui (byop-demo) 😊
0s @ 15:07:38 $
```

Custom Content Viewer

- Special content viewers can be registered
 - Image (default)
 - Video
 - Visualization



Custom Content Viewer

- Code in `nifi-format-viewer`
 - Java classes for objects & logic
 - CSS/JSP for presentation

```
1. nifi-media-bundle/nifi-image-viewer (byop-demo) 😊
└─ 3s @ 15:20:49 $ tl
  .
  └─ [2.0K] nifi-image-viewer.iml
  └─ [2.0K] pom.xml*
  └─ [ 96] src/
    └─ [ 128] main/
      └─ [ 96] java/
        └─ [ 96] org/
          └─ [ 96] apache/
            └─ [ 96] nifi/
              └─ [ 96] web/
                └─ [2.1K] ImageViewerController.java*
  └─ [ 128] webapp/
    └─ [ 96] META-INF/
      └─ [ 811] nifi-content-viewer*
  └─ [ 128] WEB-INF/
    └─ [ 96] jsp/
      └─ [1.4K] image.jsp*
    └─ [1.5K] web.xml*
11 directories, 6 files
└─ 0s @ 15:20:51 $
```

Best practices

ExecuteScript vs. InvokeScriptedProcessor

- ES easier for rapid development
- ISP more performant (should always be used in “production” flows)
 - Allows custom relationships

Module organization

- API module
 - If using controller services; separate the API and implementation into separate modules to allow use & extension
 - Can be co-located with processors for simple CS implementations

```
└── nifi-elasticsearch-nar-bundle (byop-demo) 😊
    ├── 480B Mar 12 20:36 ./nifi-elasticsearch-nar-bundle.jar
    ├── 2.4K Mar 12 20:36 .../target/nifi-elasticsearch-nar-bundle.jar
    ├── 192B Mar 12 20:39 nifi-elasticsearch-5-nar/
    ├── 192B Mar 12 20:39 nifi-elasticsearch-5-processors/
    ├── 1.3K Jan  3 14:26 nifi-elasticsearch-bundle.iml
    ├── 192B Mar 12 20:39 nifi-elasticsearch-client-service/
    ├── 192B Mar 12 20:37 nifi-elasticsearch-client-service-api/
    ├── 160B Mar 12 20:38 nifi-elasticsearch-client-service-api-nar/
    ├── 192B Mar 12 20:39 nifi-elasticsearch-client-service-nar/
    ├── 192B Mar 12 20:40 nifi-elasticsearch-nar/
    ├── 192B Mar 12 20:39 nifi-elasticsearch-processors/
    ├── 192B Mar 12 20:39 nifi-elasticsearch-restapi-nar/
    ├── 192B Mar 12 20:38 nifi-elasticsearch-restapi-processors/
    ├── 2.7K Mar  6 14:47 pom.xml
    └── 128B Mar 12 20:36 target/
```

Module configuration

- Check archetype version
- Check **nifi-api** version
- Inherits from **nifi-nar-bundle** by default
 - If building outside of NiFi, can be removed

Processor structure (con't)

- Initialize relationships and property descriptors in `init()`
 - Large allowable values collections can use builder patterns or `Enum`
- Don't connect to external services in `onScheduled()`
 - Use `onTrigger()`
- Handle dynamic properties
- Custom validation (see `StandardValidators`)

Other processor concerns

- Ensure good documentation
- Handle flowfile content in streaming manner
- Generate proper provenance events for interaction with external systems
 - Send, fetch, etc.
- Define appropriate relationships

Remember to list processors in manifest

- Most common issue when not seeing processor in NiFi
 - `src/main/resources/META-INF/services/`
 - `org.apache.nifi.processor.Processor`

```
23  org.apache.nifi.processors.standard.ConvertRecord
24  org.apache.nifi.processors.standard.CountText
25  org.apache.nifi.processors.standard.CryptographicHashAttribute
26  org.apache.nifi.processors.standard.CryptographicHashContent
27  org.apache.nifi.processors.standard.DebugFlow
28  org.apache.nifi.processors.standard.DetectDuplicate
29  org.apache.nifi.processors.standard.DistributeLoad
30  org.apache.nifi.processors.standard.DuplicateFlowFile
31  org.apache.nifi.processors.standard.EncryptContent
32  org.apache.nifi.processors.standard.EnforceOrder
33  org.apache.nifi.processors.standard.EvaluateJsonPath
34  org.apache.nifi.processors.standard.EvaluateXPath
35  org.apache.nifi.processors.standard.EvaluateXQuery
36  org.apache.nifi.processors.standard.ExecuteProcess
37  org.apache.nifi.processors.standard.ExecuteSQL
38  org.apache.nifi.processors.standard.ExecuteSQLRecord
39  org.apache.nifi.processors.standard.ExecuteStreamCommand
40  org.apache.nifi.processors.standard.ExtractGrok
41  org.apache.nifi.processors.standard.ExtractText
42  org.apache.nifi.processors.standard.FetchDistributedMapCache
43  org.apache.nifi.processors.standard.FetchFile
44  org.apache.nifi.processors.standard.FetchFTP
45  org.apache.nifi.processors.standard.FetchSFTP
46  org.apache.nifi.processors.standard.FlattenJson
47  org.apache.nifi.processors.standard.ForkRecord
48  org.apache.nifi.processors.standard.GenerateFlowFile
49  org.apache.nifi.processors.standard.GenerateTableFetch
50  org.apache.nifi.processors.standard.GetFile
51  org.apache.nifi.processors.standard.GetFTP
52  org.apache.nifi.processors.standard.GetHTTP
53  org.apache.nifi.processors.standard.GetJMSQueue
54  org.apache.nifi.processors.standard.GetJMSTopic
55  org.apache.nifi.processors.standard.GetSFTP
56  org.apache.nifi.processors.standard.HandleHttpRequest
57  org.apache.nifi.processors.standard.HandleHttpResponse
58  org.apache.nifi.processors.standard.HashAttribute
59  org.apache.nifi.processors.standard.HashContent
60  org.apache.nifi.processors.standard.IdentifyMimeType
61  org.apache.nifi.processors.standard.InvokeHTTP
62  org.apache.nifi.processors.standard.JoltTransformJSON
63  org.apache.nifi.processors.standard.ListDatabaseTables
64  org.apache.nifi.processors.standard.ListenHTTP
```

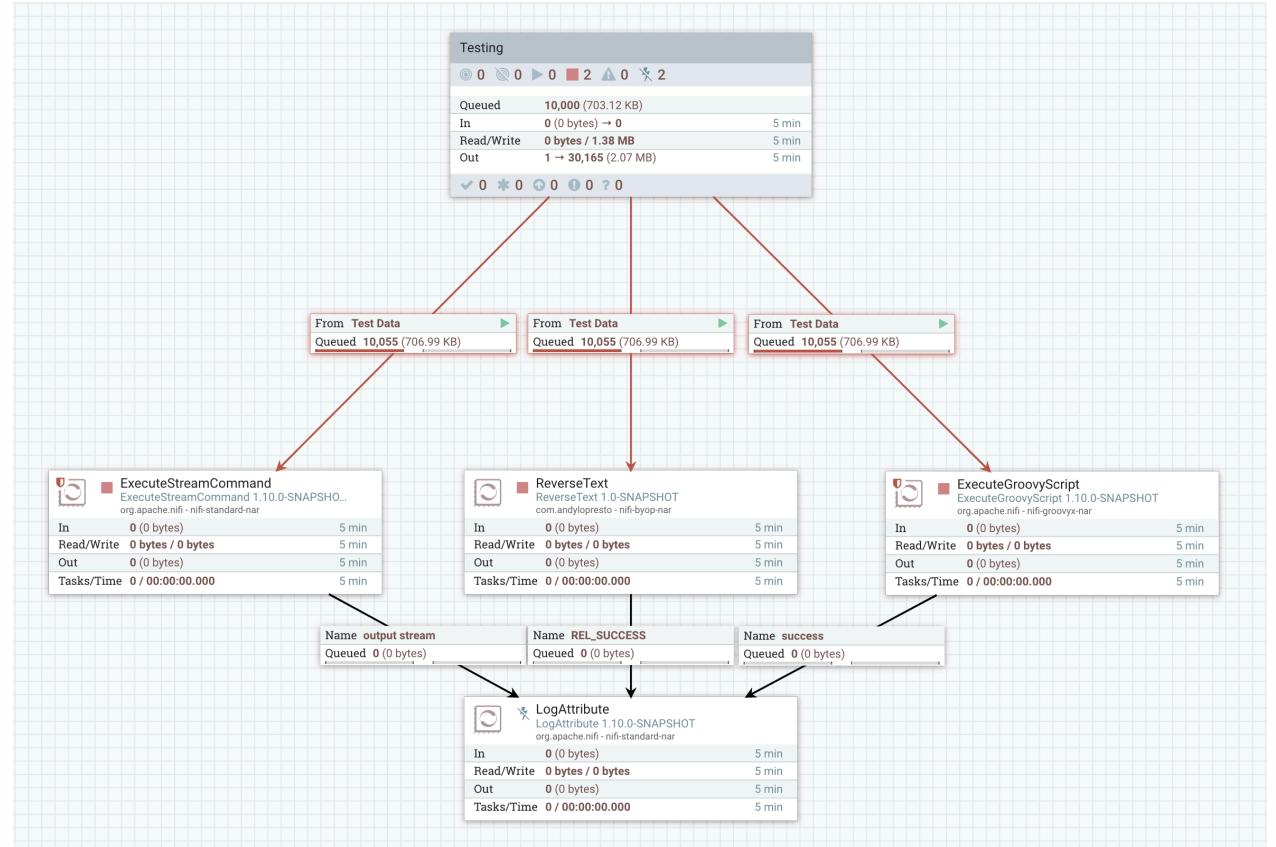
Takeaways

Dynamic vs. compiled processors

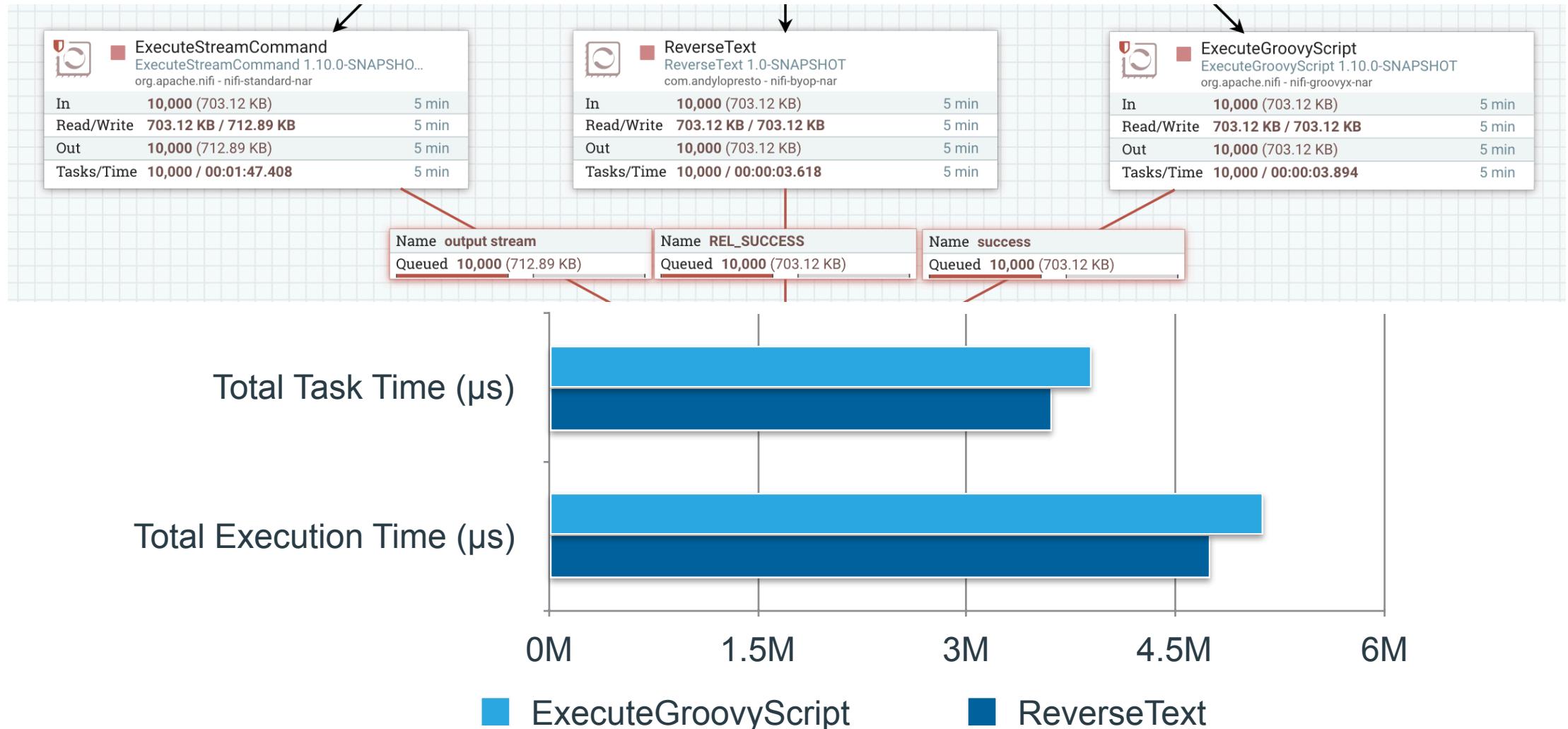
Processor type	Pros	Cons
ExecuteScript/ExecuteGroovyScript	<ul style="list-style-type: none">• Very fast development cycle• Easy to use• Source code maintained in version control if stored internally	<ul style="list-style-type: none">• Poor performance• Source code not maintained if located on file system• Limited capabilities
InvokeScriptedProcessor	<ul style="list-style-type: none">• Fast development cycle• Complex scripting capabilities• Better performance than E(G)S	<ul style="list-style-type: none">• Performance still far below compiled processor• Same limitations on code control and interaction with internal framework
Custom Processor	<ul style="list-style-type: none">• Full first-class interaction with framework API• Robust testing mechanisms• High performance• Reusability• Versioning• Ease-of-use (drag & drop)	<ul style="list-style-type: none">• Slow development cycle• Requires boilerplate code

Performance testing

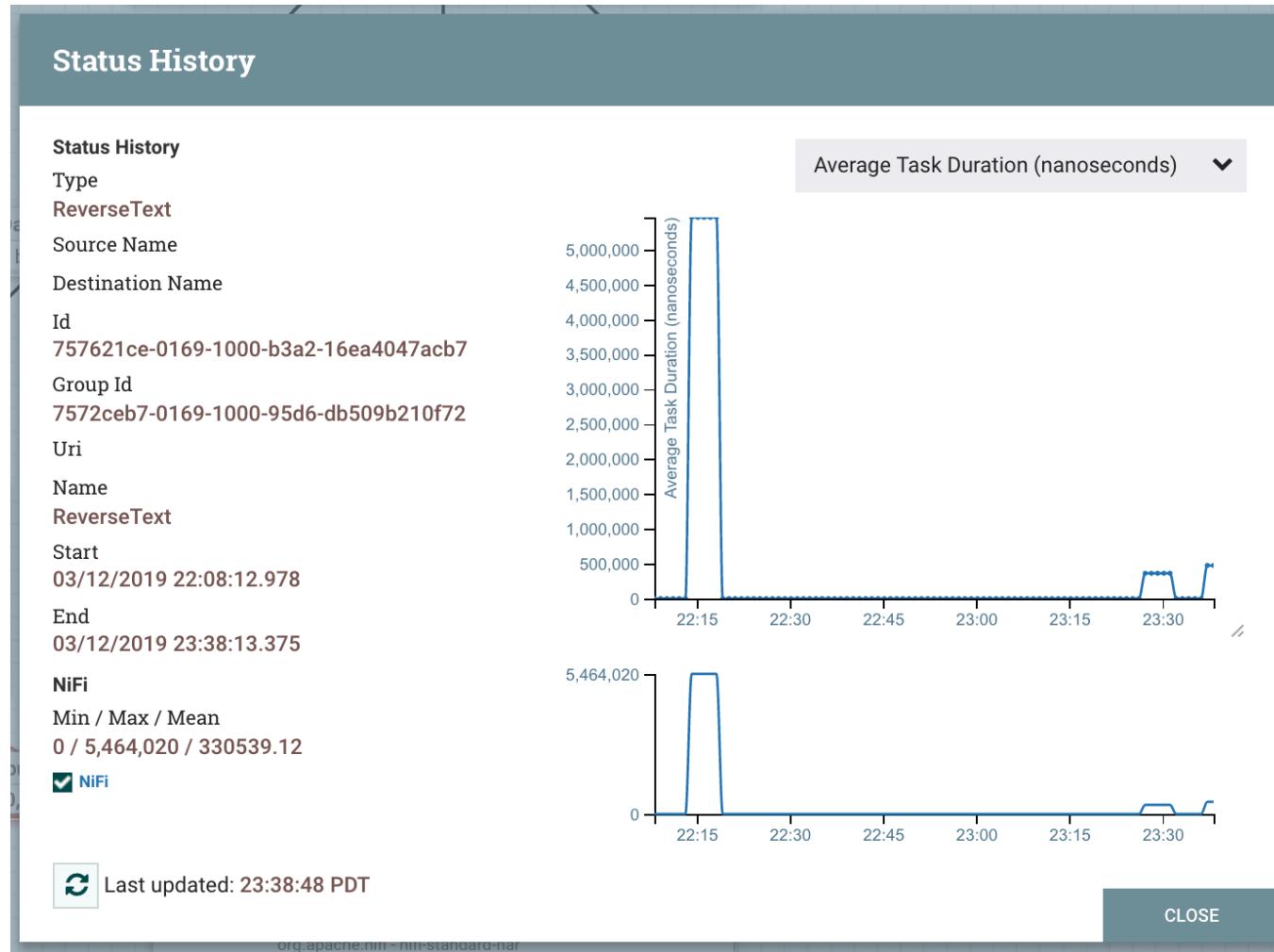
- GenerateFlowFile
- Filled back pressure of queues
- Comparing
 - STDIN -> rev -> STDOUT
 - Custom processor
 - Groovy script



Performance results



Performance results



Notes on performance

- Very simple operation (JVM optimizing)
 - External service calls, loops, etc. will have bigger impact
- Multi-threading / time-series dependent data can influence
- Streaming from file system will impact

Additional resources

Other ideas

- Build custom processor with any JVM-compatible language
 - Community accepts contributions in Java only
- Host custom processor bundles on individual repos
- Extension registry coming soon
 - PRs accepted recently

Community resources

- Apache NiFi Contributor Guide
 - Instructions on cloning repo; building; contributing code; checkstyle guide
- Apache NiFi Developer Guide
 - Processor API; lifecycle; documenting; common patterns; errors; testing
- Walkthroughs
 - Matt Burgess - funnifi.blogspot.com
 - Bryan Bende - bryanbende.com

New Announcements

- NiFi 1.9.1 – 16 Mar 2019 (167+ Jiras)
 - New processors (Kudu, Slack)
 - Hot-loading NARs
 - Security HTTP headers
 - Record processors automatically infer schema
- MiNiFi C++ 0.6.0 – Voting now...
- MiNiFi Java 0.5.0 – 7 July 2018
- NiFi Registry 0.3.0 – 25 Sept 2018



Learn more and join us

Apache NiFi site

<https://nifi.apache.org>

Subproject MiNiFi site

<https://nifi.apache.org/minifi/>

Subscribe to and collaborate at

dev@nifi.apache.org

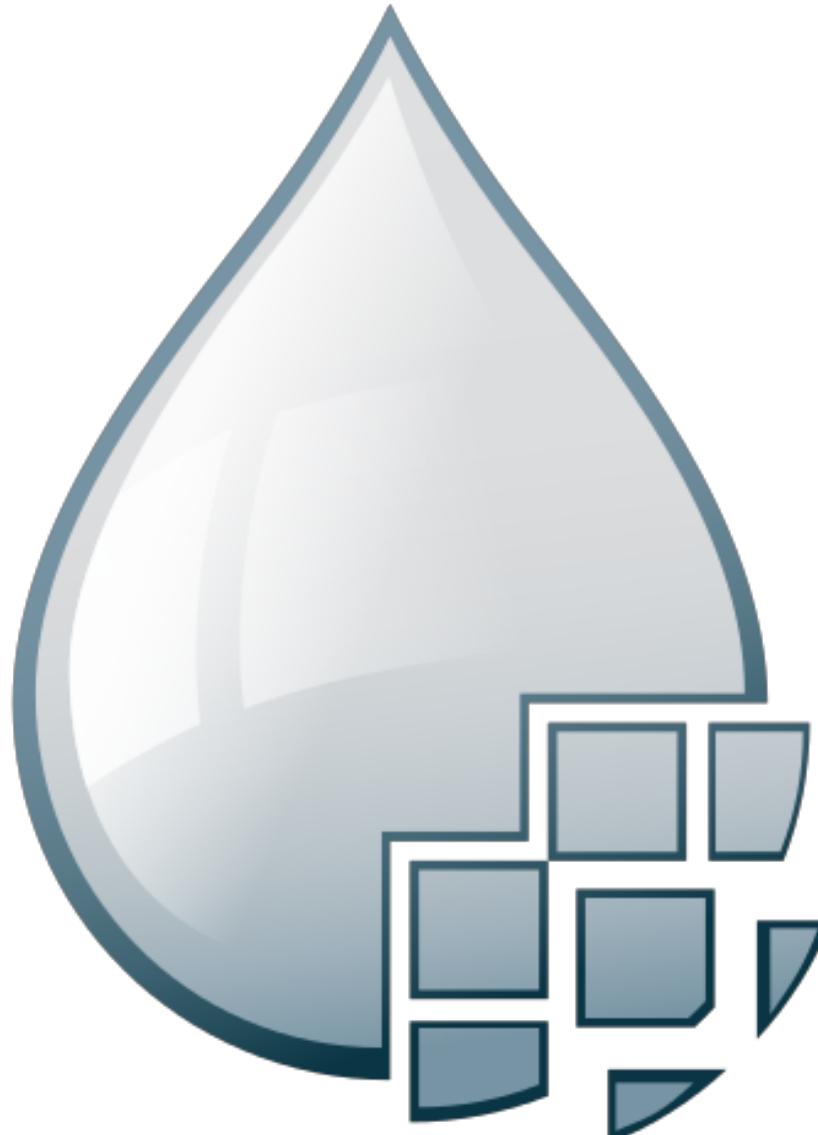
users@nifi.apache.org

Submit Ideas or Issues

<https://issues.apache.org/jira/browse/NIFI>

Follow us on Twitter

[@apachennifi](https://twitter.com/apachennifi)



NiFi at Dataworks Summit Barcelona

Title	Date	Speakers	Room	
Cloudera Data in Motion Meetup - Future of Data Barcelona (free)	Tuesday 3/19	1800 - 2000	Tim Spann, Dan Chaffelson, Andy LoPresto	127-128
Apache NiFi Crash Course		1400 - 1600	Nathan Gough, Andy LoPresto	111
Data Acquisition Automation for NiFi in a Hybrid Cloud environment – the Path towards DataOps	Wednesday 3/20	1450 - 1530	Arda Basar, Ivan Georgiev	129-130
IoT, Streaming, and Dataflow Birds of a Feather		1600 - 1730	Tim Spann, Dan Chaffelson, Purnima Reddy Kuchikulla, Andy LoPresto	129-130
Addressing Challenges with IoT Edge Management	Thursday 3/21	1100 - 1140	Dinesh Chandrasekhar	127-128
Intelligently Collecting Data at the Edge – Intro to Apache MiNiFi		1150 - 1230	Andy LoPresto	127-128
Edge to AI: Analytics from Edge to Cloud with Efficient Movement of Machine Data		1400 - 1440	Tim Spann	129-130
BYOP: Custom Processor Development with Apache NiFi		1600 - 1640	Andy LoPresto	131-132
Apache Deep Learning 201		1650 - 1730	Tim Spann	122-123
Platform for the Research and Analysis of Cybernetic Threats		1650 - 1730	Monica Franceschini	124-125

Questions

THANK YOU

alopresto@cloudera.com | alopresto@apache.org | [@yolopey](https://twitter.com/yolopey)
github.com/alopresto/slides