

# Big Data with ADAMS

## Extending ADAMS

# How to extend?

---

- From easy to hard
  - Scripting
    - Groovy
    - Jython
  - Java Project
    - simply add ADAMS jars to CLASSPATH
  - Maven Project
    - create custom *pom.xml* and build your own project
    - check out ADAMS source code and compile yourself

# API basics (2)

---

- dynamic class discovery
- Actor life cycle
  - `setUp(): String`  
configures the actor/flow, may return error message
  - `execute(): String`  
executes the actor/flow, may return error message
  - `wrapUp()`  
finishes up execution, leaves graphical output untouched
  - `cleanUp()`  
destructive, removes graphical output

**NB:** Skipping *adams.flow* package prefix in following slides

# API basics

---

- Interfaces
  - `core.Actor` - all actors
  - `core.InputConsumer` - processes input
    - `input(Token)`
  - `core.OutputProducer` - generates output
    - `hasPendingOutput(): boolean`
    - `output(): Token`
- abstract classes
  - `core.AbstractActor`
  - `standalone.AbstractStandalone`
  - `source.Abstract[Simple]Source`
  - `transformer.AbstractTransformer`
  - `sink.AbstractSink`




# API basics (3)

---

- graphical sinks
  - `sink.AbstractDisplay`
  - `sink.ComponentSupplier`
    - `sink.AbstractGraphicalDisplay`
  - `sink.TextSupplier`
    - `sink.AbstractTextualDisplay`
- interactive actors
  - Interfaces
    - `core.InteractiveActor`
    - `core.AutomatableInteractiveActor`
  - abstract classes
    - `source.AbstractInteractiveSource`
    - `transformer.AbstractInteractiveTransformer`



# API basics (4)

---

- Actor handlers manage data flow
  - interface: `control.ActorHandler`
  - some use “directors” to direct the data stream, e.g.:
    -  Sequence
    -  IfThenElse
    -  Switch









# Scripting

---

- Available through these modules
  -  adams-groovy
  -  adams-jython
- Advantages
  - no compilation required
  - access to all libraries on CLASSPATH
  - fast prototyping
- Limitations
  - code executed in same JVM
  - only pure-Python code will execute (no numpy!)

# Scripting

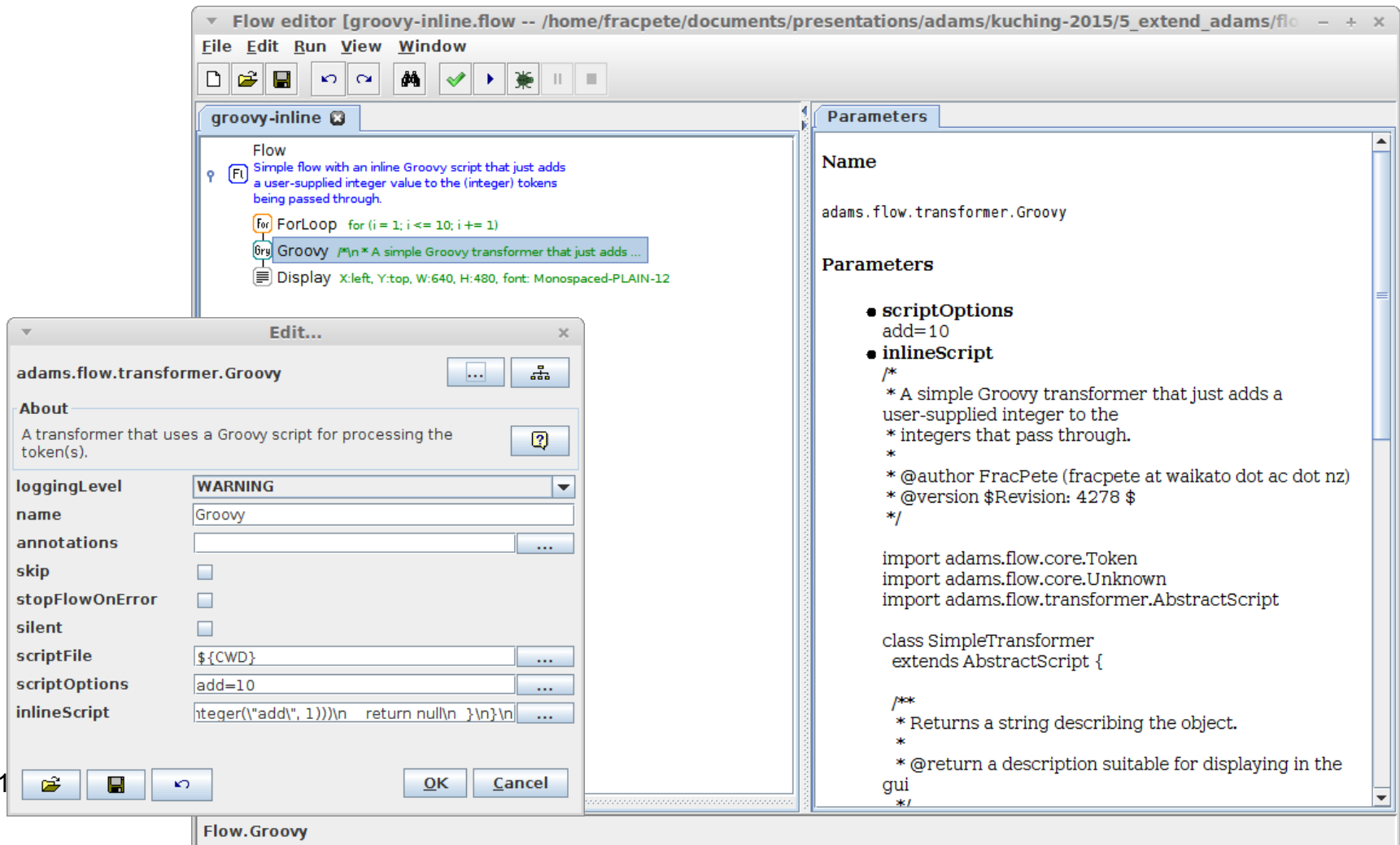
---

- Actors available
  -   Standalones
  -   Sources
  -   Transformers
  -   Sink



# Scripting - Groovy

- Inline script



The screenshot shows the Flow editor interface. The main window displays a flow named "groovy-inline" with the following steps:

- Flow: Simple flow with an inline Groovy script that just adds a user-supplied integer value to the (integer) tokens being passed through.
- ForLoop: for (i = 1; i <= 10; i += 1)
- Groovy: /\*\n \* A simple Groovy transformer that just adds ...
- Display: X:left, Y:top, W:640, H:480, font: Monospaced-PLAIN-12

An "Edit..." dialog box is open, showing the configuration for the "adams.flow.transformer.Groovy" transformer. The configuration includes:

- About:** A transformer that uses a Groovy script for processing the token(s).
- loggingLevel:** WARNING
- name:** Groovy
- annotations:** (empty)
- skip:** ☐
- stopFlowOnError:** ☐
- silent:** ☐
- scriptFile:** \${CWD}
- scriptOptions:** add=10
- inlineScript:** integer("add", 1))\n return null\n }\n}\n

The "Parameters" tab on the right shows the following details:

- Name:** adams.flow.transformer.Groovy
- Parameters:**
  - scriptOptions:** add=10
  - inlineScript:**

```
/*
 * A simple Groovy transformer that just adds a
 * user-supplied integer to the
 * integers that pass through.
 *
 * @author FracPete (fracpete at waikato dot ac dot nz)
 * @version $Revision: 4278 $
 */

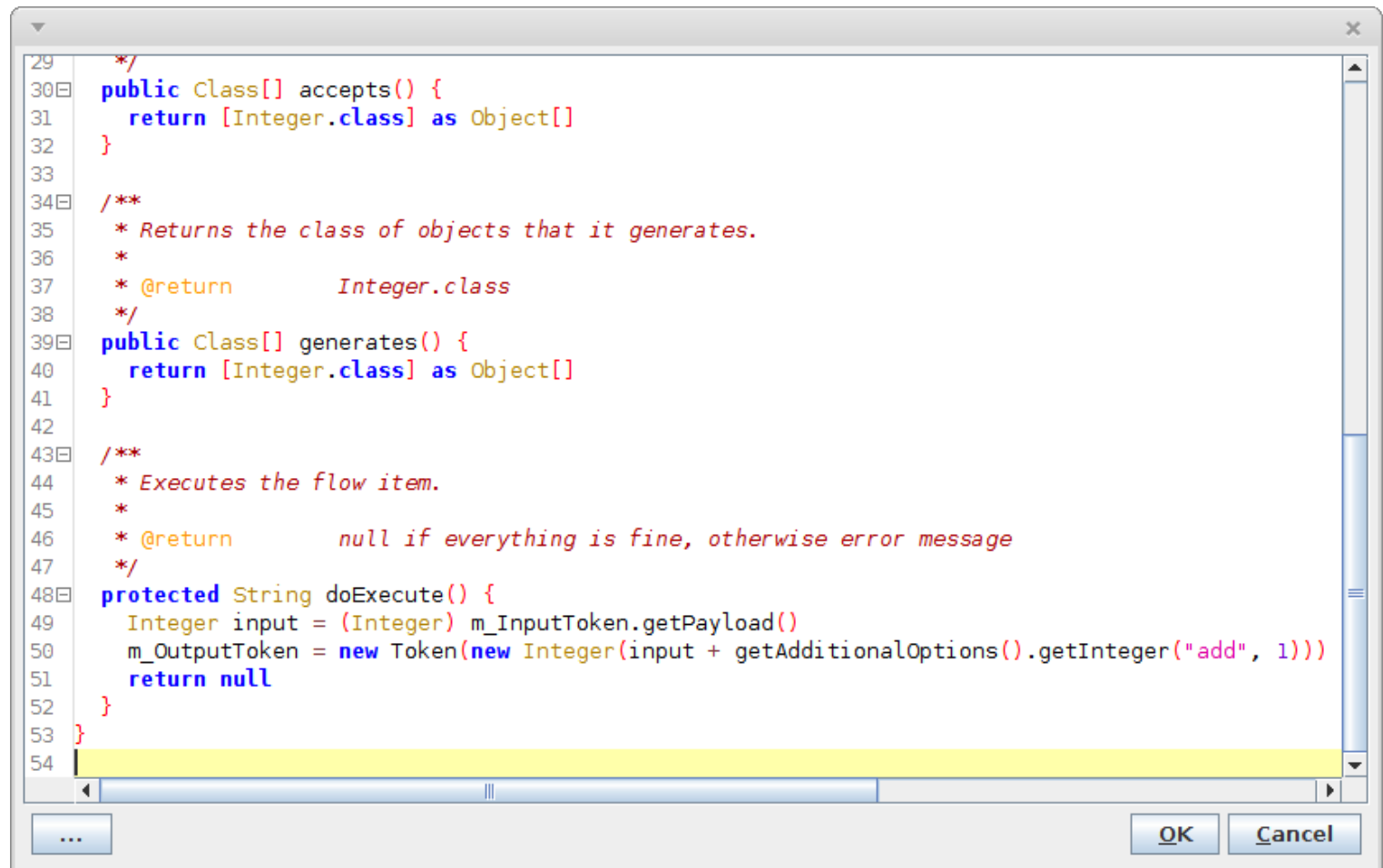
import adams.flow.core.Token
import adams.flow.core.Unknown
import adams.flow.transformer.AbstractScript

class SimpleTransformer
extends AbstractScript {

/**
 * Returns a string describing the object.
 *
 * @return a description suitable for displaying in the
 * gui
 */
```

# Scripting - Groovy

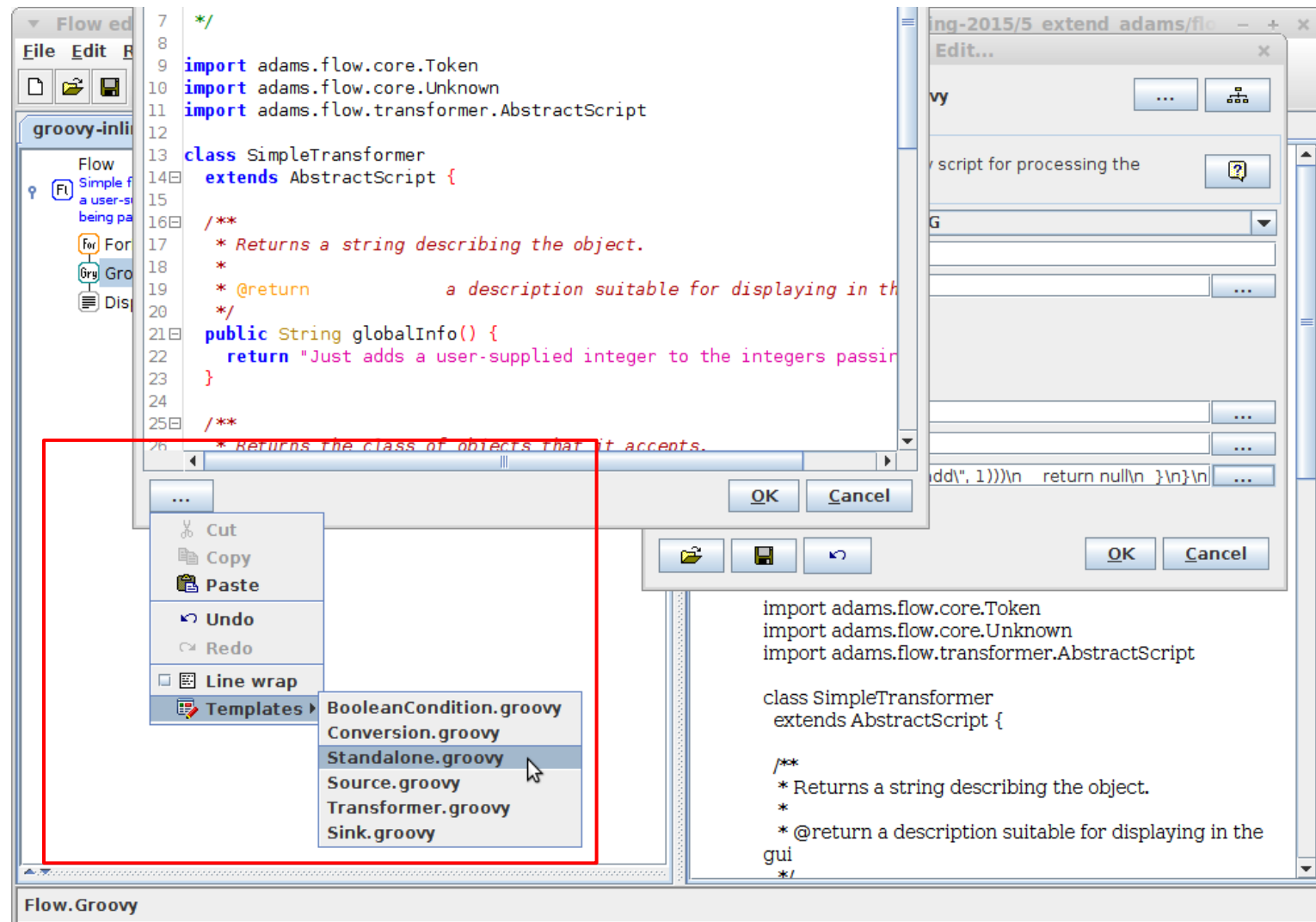
- Inline editor



```
29  */
30  public Class[] accepts() {
31      return [Integer.class] as Object[]
32  }
33
34  /**
35   * Returns the class of objects that it generates.
36   *
37   * @return      Integer.class
38   */
39  public Class[] generates() {
40      return [Integer.class] as Object[]
41  }
42
43  /**
44   * Executes the flow item.
45   *
46   * @return      null if everything is fine, otherwise error message
47   */
48  protected String doExecute() {
49      Integer input = (Integer) m_InputToken.getPayload()
50      m_OutputToken = new Token(new Integer(input + getAdditionalOptions().getInteger("add", 1)))
51      return null
52  }
53  }
54
```

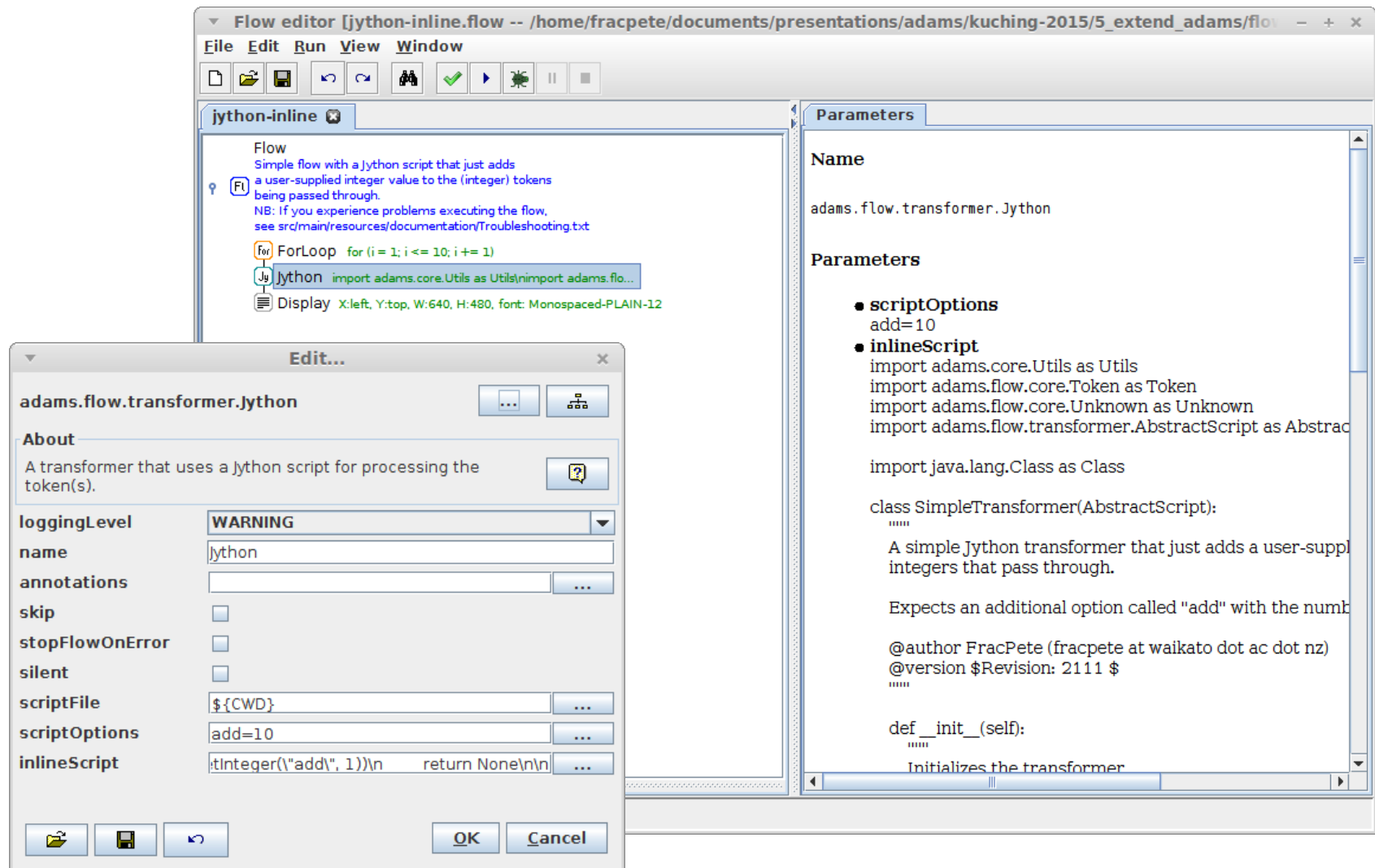
# Scripting - Groovy

- Templates



# Scripting - Jython

- Inline script



The screenshot displays the Flow editor interface for a Jython inline script. The main window shows a flow diagram with a 'Jython' node. The 'Parameters' tab on the right lists the script options and the inline script code.

**Flow editor [jython-inline.flow -- /home/fracpete/documents/presentations/adams/kuching-2015/5\_extend\_adams/flow -- + x]**

**File Edit Run View Window**

**jython-inline**

Flow  
Simple flow with a Jython script that just adds a user-supplied integer value to the (integer) tokens being passed through.  
NB: If you experience problems executing the flow, see [src/main/resources/documentation/Troubleshooting.txt](#)

**ForLoop** for (i = 1; i <= 10; i += 1)

**Jython** import adams.core.Utils as Utils; import adams.flow...  
Display X:left, Y:top, W:640, H:480, font: Monospaced-PLAIN-12

**Parameters**

**Name**  
adams.flow.transformer.Jython

**Parameters**

- **scriptOptions**  
add=10
- **inlineScript**  
import adams.core.Utils as Utils  
import adams.flow.core.Token as Token  
import adams.flow.core.Unknown as Unknown  
import adams.flow.transformer.AbstractScript as AbstractScript  
  
import java.lang.Class as Class  
  
class SimpleTransformer(AbstractScript):  
 """  
 A simple Jython transformer that just adds a user-supplied integers that pass through.  
  
 Expects an additional option called "add" with the number  
  
 @author FracPete (fracpete at waikato dot ac dot nz)  
 @version \$Revision: 2111 \$  
 """  
  
 def \_\_init\_\_(self):  
 """  
 Initializes the transformer  
 """

**Edit...**

**adams.flow.transformer.Jython**

**About**  
A transformer that uses a Jython script for processing the token(s).

**loggingLevel** WARNING

**name** jython

**annotations**

**skip** ☐

**stopFlowOnError** ☐

**silent** ☐

**scriptFile** \${CWD}

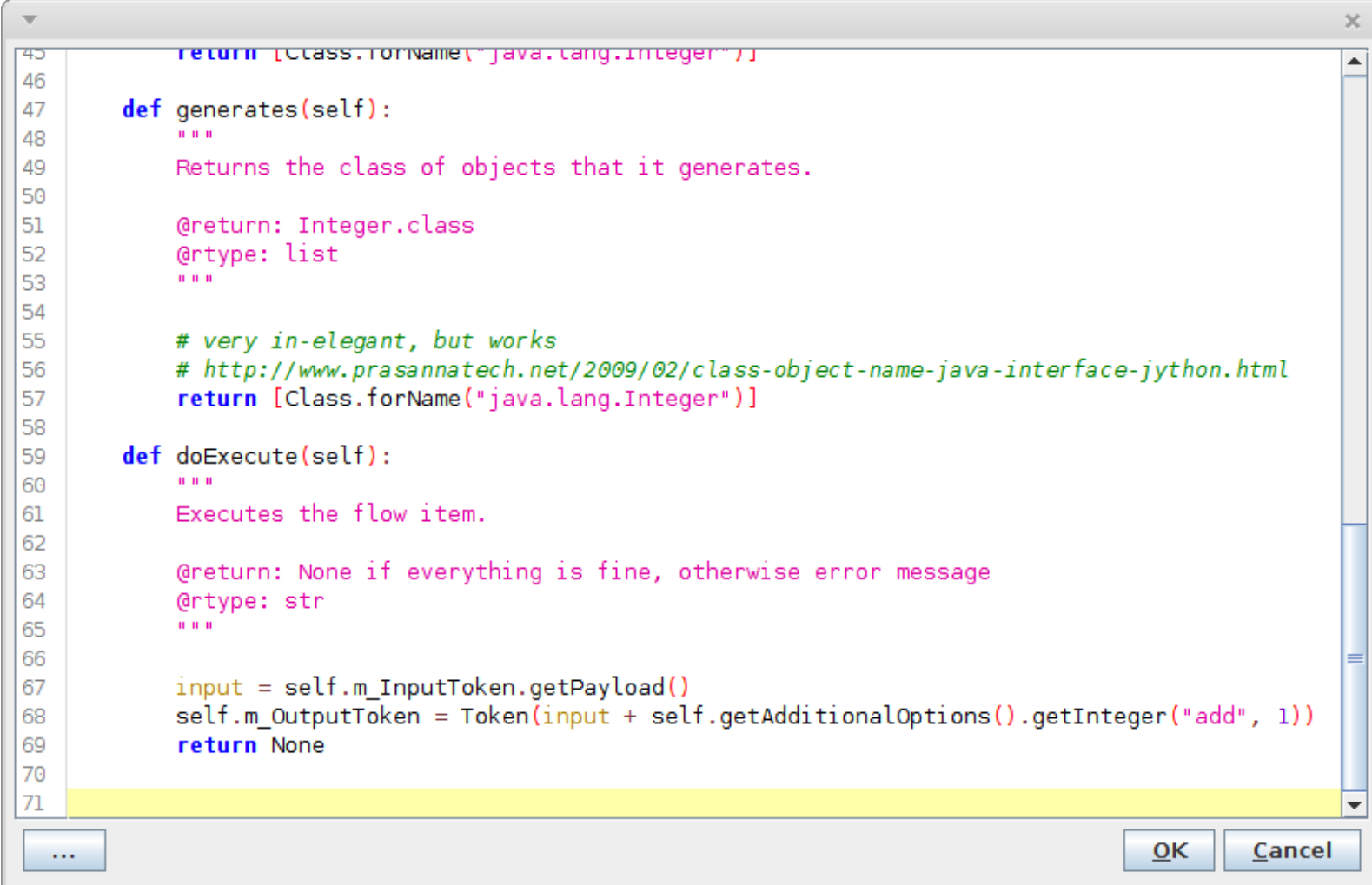
**scriptOptions** add=10

**inlineScript** t.Integer("add", 1))\n return None\n\n

OK Cancel

# Scripting - Jython

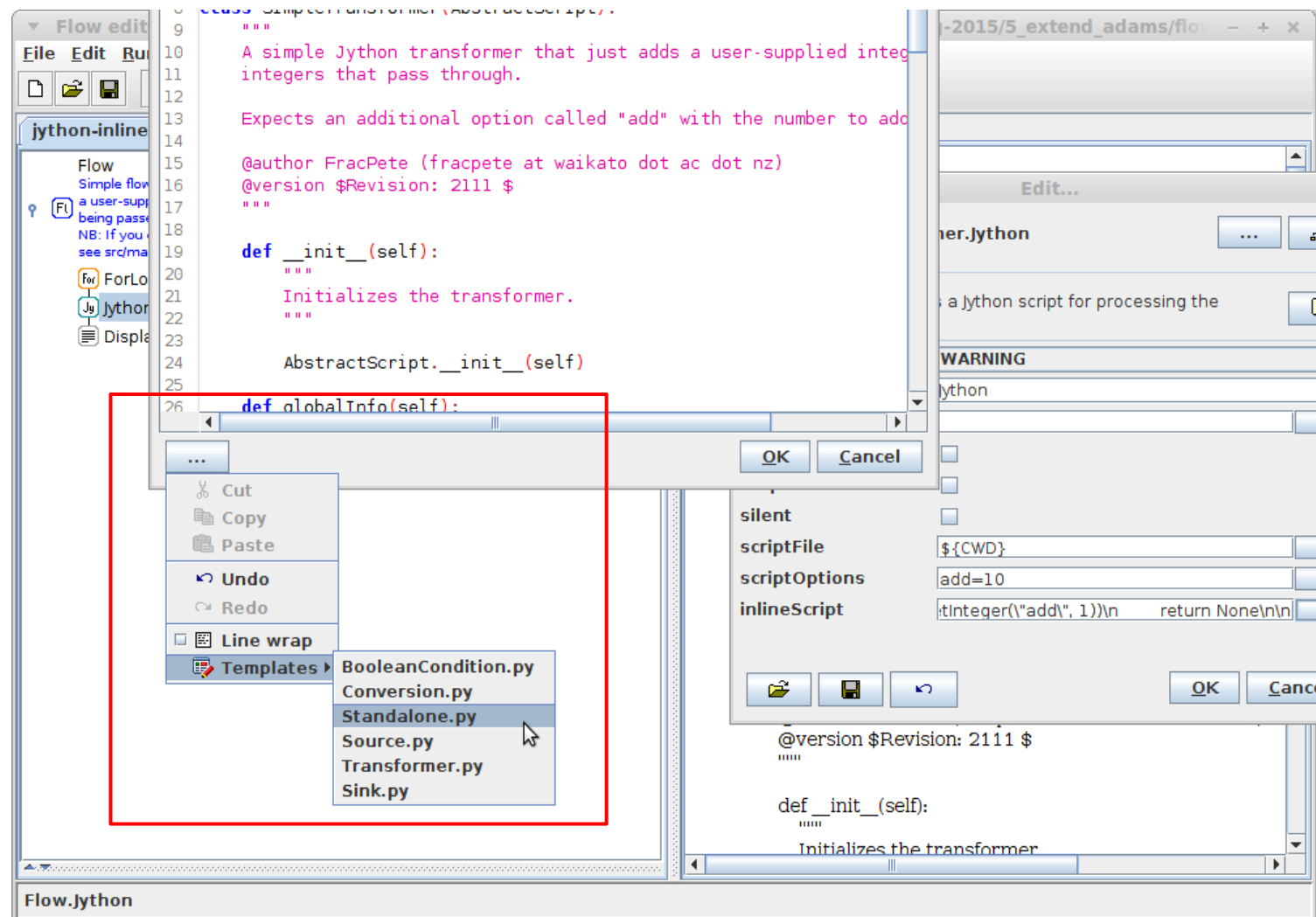
- Inline editor



```
45         return [Class.forName("java.lang.Integer")]
46
47     def generates(self):
48         """
49         Returns the class of objects that it generates.
50
51         @return: Integer.class
52         @rtype: list
53         """
54
55         # very in-elegant, but works
56         # http://www.prasannatech.net/2009/02/class-object-name-java-interface-jython.html
57         return [Class.forName("java.lang.Integer")]
58
59     def doExecute(self):
60         """
61         Executes the flow item.
62
63         @return: None if everything is fine, otherwise error message
64         @rtype: str
65         """
66
67         input = self.m_InputToken.getPayload()
68         self.m_OutputToken = Token(input + self.getAdditionalOptions().getInteger("add", 1))
69         return None
70
71
```

# Scripting - Jython

- Templates



# Java Project

---

- Download release or snapshot, decompress
- Start your favorite IDE
- Add jars from “lib” directory to the project's CLASSPATH
- Ensure that “java-cup-11b-2015.03.26.jar” is listed first
- Done!

# Maven Project

---

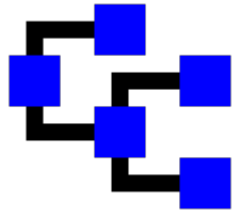
- Requires Maven 3.0+  
<http://maven.apache.org/>
- Two options
  - use “roll your own” feature, use existing artifacts \*  
<https://adams.cms.waikato.ac.nz/rollyourown.html>
  - check out from subversion, compile yourself  
[https://adams.cms.waikato.ac.nz/dev\\_start.html](https://adams.cms.waikato.ac.nz/dev_start.html)  
**NB:** also requires LaTeX installed

\* *Maven needs to use the ADAMS Nexus server as mirror ([settings.xml](#))*



# Questions?

---



<https://adams.cms.waikato.ac.nz/>

@TheAdamsFlow