

4 - Modifying a workflow

This tutorial tests your understanding of the previous one ([Tutorial 3 - Defining Workflows in YAML](#)) by asking you to modify a workflow to incorporate additional functionality. The goal will be update `hello.yaml` to make the emphasis in the greeting (the exclamation point in the output of `hello.yaml`) configurable by the user of the workflow.

1. Start from the contents of `hello.yaml`

Copy the contents of `hello.yaml` (or copy from the workflow definition immediately below) into a new text file and save it with a new name, e.g. `hello_emphasized.yaml`. Your file should now contain the following.

```
imports:

- classpath:/org/kurator/akka/actors.yaml

components:

- id: GreetingSource
  type: ConstantSourceActor
  properties:
    parameters:
      value: Hello World!

- id: GreetingPrinter
  type: PrinterActor
  properties:
    listensTo:
      - !ref GreetingSource

- id: HelloWorldWorkflow
  type: Workflow
  properties:
    actors:
      - !ref GreetingSource
      - !ref GreetingPrinter
    parameters:
      greeting:
        actor: !ref GreetingSource
        parameter: value
```

2. Remove the emphasis from the default greeting

The first thing to do is to remove the exclamation point from the default value for `GreetingSource`. Update `GreetingSource` so that default value assigned to the value parameter is simply 'Hello World' (the single quotes are optional).

3. Add a new actor for appending emphasis

Kurator-Akka includes a `StringAppender` actor that adds a configurable suffix to each string it receives and outputs the modified strings. Add a new component to your workflow file (it may be easiest to copy and paste the `GreetingSource` component). It doesn't matter where you add the component, as long as it is somewhere in the components section of the file. Give the new component an id of `GreetingEmphasizer` and a type of `StringAppender`. `StringAppender` actors take a parameter called `suffix`. Edit your `GreetingEmphasizer` component to make a single exclamation point the default suffix (enclose the exclamation point in single quotes). Finally, add a `listensTo` property to `GreetingPrinter` and configure the component to receive its input from `GreetingSource`.

4. Reconfigure `GreetingPrinter` to receive the emphasized greeting

In `hello.yaml`, `GreetingPrinter` receives the string that it prints directly from `GreetingSource`. Modify the configuration for `GreetingPrinter` so that it now receives the emphasized greeting from `GreetingEmphasizer` instead.

5. Configure the workflow component to include `GreetingEmphasizer`

The workflow component includes a list of actors to be included in the workflow. Add a reference to `GreetingEmphasizer` to that list.

6. Make the emphasis added by `GreetingEmphasizer` configurable by the user

The workflow component in `hello.yaml` maps the value parameter of `GreetingSource` to the greeting parameter of the workflow as a whole so that users can override the greeting. Update the `HelloWorldWorkflow` component in the modified workflow to also expose the suffix parameter of the `GreetingEmphasizer` actor as a workflow parameter named `emphasis`.

7. Try running your modified workflow

First make sure that your workflow produces the same output as the original when no parameter values are overridden:

```
$ curator-akka -f hello_emphasized.yaml
Hello World!
$
```

Now trying assigning a new greeting while leaving the emphasis alone. For example,

```
$ curator-akka -f hello_emphasized.yaml -p greeting=Goodbye
Goodybe!
$
```

Finally, try customizing both the greeting and the emphasis:

```
$ curator-akka -f hello_emphasized.yaml -p greeting=Goodbye -p emphasis=.
Goodbye.
$
```

Note that the values assigned to parameters often need to be quoted to prevent shell expansion. Because Kurator-Akka interprets parameter values as YAML (or JSON) strings, so that lists and mappings may be provided as parameter values, quotes can be required to prevent interpretation as YAML as well. Using a pair of double quotes nested within a pair of single quotes can be the safest approach. For example, to use a question mark as the emphasis you will get an error unless you quote the ? in this manner (? surrounded by both double and single quotes):

```
$ curator-akka -f hello_emphasized.yaml -p greeting=Goodbye -p emphasis='"? "'
Goodbye?
$
```

8. Check your work

If your workflow does not behave as expected, compare it to the workflow below. Differences between `hello_emphasized.yaml` and `hello.yaml` are in bold.

```
imports:

- classpath:/org/kurator/akka/actors.yaml

components:

- id: GreetingSource
  type: ConstantSourceActor
  properties:
    parameters:
      value: Hello World

- id: GreetingEmphasizer
  type: StringAppender
  properties:
    parameters:
      suffix: '!'
    listensTo:
      - !ref GreetingSource

- id: GreetingPrinter
  type: PrinterActor
  properties:
    listensTo:
      - !ref GreetingEmphasizer

- id: HelloWorldWorkflow
  type: Workflow
  properties:
    actors:
      - !ref GreetingSource
      - !ref GreetingEmphasizer
      - !ref GreetingPrinter
    parameters:
      greeting:
        actor: !ref GreetingSource
        parameter: value
      emphasis:
        actor: !ref GreetingEmphasizer
        parameter: suffix
```