Continuuity BigFlow Developer Edition

BigFlow Developer Edition is a fully functioning stand alone edition of the Continuuity BigFlow platform. You can use this edition to write, debug and test your BigData applications, before deploying to the Continuuity Cloud Platform. This document contains the following sections:

1. **Installation** - pre-requisite and installation instructions
2. **Directory Overview** - a brief discussion of the directories included in the product
3. **Tools** - a brief overview of the tools included

To install and use BigFlow and the included examples there are a few simple pre-requisite requirements:

1. Java 1.6+ (required to run BigFlow, note: $JAVA_HOME should be set correctly)
2. Node.js 0.8.1+ (required to run the BigFlow UI)
3. Ant 1.8+ (required to build the example applications)

Once you have installed the prerequisites, you should unzip the bigflow distribution to a suitable directory. E.g.

```
% cd ~/Projects
% unzip /tmp/bigflow-developer-edition-1.0.0-alpha.zip
```

You are now ready to start BigFlow. This can be achieved by running the bigflow script in the $BIGFLOW_DIR/bin folder. E.g.

```
% cd ~/Projects/bigflow-developer-edition-1.0.0-alpha
% bin/bigFlow
```

The directory structure for BigFlow is shown below.

```
bigflow-developer-edition-1.0.0-alpha
|
|-- README (this file)
|-- LICENSE
|-- continuuity-api.jar (the API jar for BigFlow)
|-- bin (execution/tools scripts)
| |-- bigFlow
| |-- bigFlow.bat
| |-- data-format
| |-- stream-client
| |-- data-client
| |-- flow-client
|-- lib
|-- docs
| |-- api (api javadocs)
|-- conf (configuration files)
| |-- logback.xml
| |-- continuuity-webapp.xml
| |-- continuuity-overlord.xml
| |-- continuuity-gateway.xml
| |-- continuuity-flow.xm
| |-- continuuity-data-fabric.xml
|-- web-app (BigFlow UI application)
|-- examples (example applications)
| |-- CountTokens
| |-- CountRandom
| |-- build.xml
```

BigFlow comes with a suite of cmdline tools that allow you to perform certain functions such as uploading flows, reading or writing from the data fabric or send events to an Event Stream. The list of tools is outlined below:

Flow Client

The flow client is a command line utility to deploy and manage flows. To deploy a flow:

```
flow-client --command=deploy --resource=myflow.jar
```

After it is deployed, you can start, stop, or delete it, using the application name and flow name of the flow:

```
flow-client --command=start --application=myapp --flow=myflow
flow-client --command=stop --application=myapp --flow=myflow
flow-client --command=delete --application=myapp --flow=myflow
```

Make sure to use the same application name and flow name that was specified in the flow when it was deployed.

Stream Client

The stream client is a utility to send events to a stream, or to view the contents of a stream. Note that it only operates on event streams (but not on intra-flow queues, which contain tuples).

To send an event to a stream, invoke `stream-client` with the `send` command. For instance:

```
stream-client send --stream myflow --header number "10" --body "This is message number 10"
```

sends and event to the stream named myflow with one header and a text body. Also:

- You can add multiple headers by repeating the `--header` option
- You can use hexadecimal notation or URL-encoding to specify the body of the event on command line (default is URL-encoding)
- You can read the body from a binary file if it that is more convenient than encoding it into a printable string.

You can also use `stream-client` to inspect the contents of a stream using the `view` command. For instance

```
stream-client view --stream myflow --last 50
```

prints the last 50 events that were added to the stream named `myflow`. Similarly to `send`:

- You can use hexadecimal or URL-encoding to format the body of each event as a printable string (default is URL-encoding).
- You can influence what parts of the stream's content to view by specifying using `--all`, `--first`, or `--last`

Note that the stream client is a convenience wrapper around BigFlow's REST interface, and you can achieve the same functionality by making HTTP calls.

A note on streams: To send events to a flow, you typically use the flow name as the stream name. Stream client will then send the event to the flow's default input stream. If you defined custom input streams for the flow, then you can use *flowName/streamName* to address it.

Data Client

The data client is a command line utility to access the data persisted in the data fabric by flows. It supports reading, writing or deleting the value for a key, listing all keys, and formatting the data fabric.

For instance, to see the value for a given key:

```
data-client read --key number-42
```

will print the value stored for the key `number-42` to the console (or print "Not Found" if there is no value stored for that key). Because in the data fabric all keys and value are binary byte-arrays, data client converts binary strings to and from printable strings using the default system encoding. You can change that using `--url` for URL-encoding and `--hex` for hexadecimal encoding. This applies both to the key and the value, that is, if you specify the key in hexadecimal notation, then the value will be printed in the same notation.

If you determine that a flow wrote a bad value and wish to delete it, you can do so using the `delete` command:

```
data-client delete --key badKey
```

This will delete the value for the given key, and succeeding read will not see a value. In order to change the value for a key, use the `write` command:

```
data-client write --key 42 --value "what is the answer?"
```

As always, use `--url` or `--hex` to change the encoding used for converting to/from binary strings.

If you are not sure what keys exist in the data fabric, you can list all keys by "paging" over the key range:

```
data-client list --limit 50
data-client list --limit 50 --start 50
data-client list --limit 50 --start 100
...
```

Finally, if you want to clear all data stored in the data fabric, you can use the `format` command:

```
data-client format --data
```

will delete all key/value pairs stored in the fabric. You can also specify `--streams` to clear all event streams, `--queues` to delete all intra-flow queues, or `--all` to delete all key/value data, streams and queues at once. Caution: Once you format your data fabric, its contents cannot be restored.

BigFlow UI

In addition, BigFlow Developer Edition comes with a bundled web based user interface (UI) that allows you to deploy and manage your Flow applications. After you have started BigFlow successfully you can connect to the UI at http://localhost:9999 (unless you have changed the default port for the UI)