

# Scripting NiFi Data Flows

---

## Scripting NiFi Data Flows

### Recipes

[Writing Log Message](#)

[Read From FlowFile](#)

[Write to a FlowFile](#)

[Overwrite a FlowFile](#)

[Example: Reverse Columns Order in CSV](#)

[Example: Convert CSV to JSON](#)

[Example: Hash Field Values with MD5](#)

[See Also](#)

[Example: Append A Column to CSV FlowFile](#)

[Usage](#)

[AddOn: Convert Bytes to Hex \(Java\)](#)

### Resources

## Recipes

---

### Writing Log Message

```
var objectArrayType = Java.type("java.lang.Object[]");
var objArray = new objectArrayType(3);
objArray[0] = 'Hello';
objArray[1] = 1;
objArray[2] = true;
log.error('Found these things: {} {} {}'.format(objArray[0], objArray[1], objArray[2]))
```

### Read From FlowFile

```

var flowFile = session.get();
if(flowFile != null) {
    // Create a new InputStreamCallback, passing in a function to define the interface
    // method
    session.read(flowFile,
        new InputStreamCallback(function(inputStream) {
            var text = IOUtils.toString(inputStream, StandardCharsets.UTF_8);
            // Do something with text here
            log.debug("Read: " + text)
        }));
}

```

## Write to a FlowFile

```

var OutputStreamCallback =
    Java.type("org.apache.nifi.processor.io.OutputStreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");

var flowFile = session.get();
if(flowFile != null) {
    // Create a new OutputStreamCallback, passing in a function to define the interface
    // method
    flowFile = session.write(flowFile,
        new OutputStreamCallback(function(outputStream) {
            outputStream.write("Hello World!".getBytes(StandardCharsets.UTF_8))
        }));
}

```

## Overwrite a FlowFile

```

var StreamCallback = Java.type("org.apache.nifi.processor.io.StreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");

var flowFile = session.get();
if(flowFile != null) {
    // Create a new StreamCallback, passing in a function to define the interface method
    flowFile = session.write(flowFile,
        new StreamCallback(function(inputStream, outputStream) {
            var text = IOUtils.toString(inputStream, StandardCharsets.UTF_8)

            outputStream.write(text.split(",").reverse().join("").getBytes(StandardCharsets.UTF_8))
        }));
}

```

## Example: Reverse Columns Order in CSV

```
var StreamCallback = Java.type("org.apache.nifi.processor.io.StreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");

var flowFile = session.get();
while (flowFile != null) {
    // Create a new StreamCallback, passing in a function to define the interface method
    flowFile = session.write(flowFile,
        new StreamCallback(function(inputStream, outputStream) {
            var text = IOUtils.toString(inputStream, StandardCharsets.UTF_8)
            var lines = text.split("\n")

            outputStream.write(
                lines.map(function(line) {
                    return line.split(",").reverse().join(",")
                }).join("\n").getBytes(StandardCharsets.UTF_8))
        }));
    session.transfer(flowFile, REL_SUCCESS);
    flowFile = session.get()
}
```

## Example: Convert CSV to JSON

```
var StreamCallback = Java.type("org.apache.nifi.processor.io.StreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");

var flowFile = session.get();
while (flowFile != null) {
    flowFile = session.write(flowFile,
        new StreamCallback(function(inputStream, outputStream) {
            var text = IOUtils.toString(inputStream, StandardCharsets.UTF_8)
            var lines = text.split("\n").filter(function(line) { return line.length > 0; })

            var fields = lines.shift().split(",");

            var transformed = lines.map(function(line) {
                var result = {};
                var values = line.split(",")
                for (var i=0; i<fields.length; i++) {
                    result[fields[i]] = values[i]
                }
                return result;
            })
        })
    )
}
```

```

        outputStream.write(
            JSON.stringify(transformed).getBytes(StandardCharsets.UTF_8)
        )
    });
    session.transfer(flowFile, REL_SUCCESS);
    flowFile = session.get()
}

```

- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/JSON/stringify](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify)

## Example: Hash Field Values with MD5

```

var StreamCallback = Java.type("org.apache.nifi.processor.io.StreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");
var MessageDigest = Java.type("java.security.MessageDigest");
var DatatypeConverter = Java.type("javax.xml.bind.DatatypeConverter");

var cryptMD5 = MessageDigest.getInstance('MD5')
var flowFile = session.get();

function md5(msg) {
    return DatatypeConverter.printHexBinary(
        cryptMD5.digest(msg.getBytes(StandardCharsets.UTF_8))
    );
}

while (flowFile != null) {
    flowFile = session.write(flowFile,
        new StreamCallback(function(inputStream, outputStream) {
            var text = IOUtils.toString(inputStream, StandardCharsets.UTF_8)
            var lines = text.split("\n").filter(function(line) { return line.length > 0; }
        )

            var fields = lines.shift().split(",");

            var transformed = lines.map(function(line) {
                var result = {};
                var values = line.split(",")
                for (var i=0; i<fields.length; i++) {
                    result[fields[i]] = md5(values[i])
                }
                return result;
            })

            outputStream.write(
                JSON.stringify(transformed).getBytes(StandardCharsets.UTF_8)
            )
        })
    )
}

```

```

    });
    session.transfer(flowFile, REL_SUCCESS);
    flowFile = session.get()
}

```

## See Also

- <http://www.dev-garden.org/2013/04/16/java-byte-array-as-a-hex-string-the-easy-way/>
- <https://stackoverflow.com/questions/9655181/how-to-convert-a-byte-array-to-a-hex-string-in-java>
- <https://stackoverflow.com/questions/4895523/java-string-to-sha1>
- 

## Using eval()

```

var StreamCallback = Java.type("org.apache.nifi.processor.io.StreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");
var MessageDigest = Java.type("java.security.MessageDigest");
var DatatypeConverter = Java.type("javax.xml.bind.DatatypeConverter");

var cryptMD5 = MessageDigest.getInstance('MD5')
var flowFile = session.get();

function md5(msg) {
    return DatatypeConverter.printHexBinary(
        cryptMD5.digest(msg.getBytes(StandardCharsets.UTF_8))
    );
}

while (flowFile != null) {
    flowFile = session.write(flowFile,
        new StreamCallback(function(inputStream, outputStream) {
            var text = IOUtils.toString(inputStream, StandardCharsets.UTF_8)
            var lines = text.split("\n").filter(function(line) { return line.length > 0; }
        )

            var fields = lines.shift().split(",");

            var transformed = lines.map(function(line) {
                var values = line.split(",")
                eval(applyExpression)
                return values;
            })

            outputStream.write(
                JSON.stringify(transformed).getBytes(StandardCharsets.UTF_8)
            )
        }));
    session.transfer(flowFile, REL_SUCCESS);
}

```

```
    flowFile = session.get()
}
```

## Example: Append A Column to CSV FlowFile

### Usage

- Use ExecuteScript NiFi processor
- Set `Script Engine` to `ECMAScript`
- Add `ColumnNameToAppend` processor property. For example: `ColumnNameToAppend = filename`
- Add `ColumnValueToAppend` processor property. This property supports Expression Language. E.g. `ColumnValueToAppend = ${filename}`
- Paste following code into the `Script Body` property.

```
var StreamCallback = Java.type("org.apache.nifi.processor.io.StreamCallback");
var IOUtils = Java.type("org.apache.commons.io.IOUtils");
var StandardCharsets = Java.type("java.nio.charset.StandardCharsets");

if (ColumnValueToAppend === undefined) {
    throw "Expected property [ColumnValueToAppend] not defined."
}

if (ColumnNameToAppend === undefined) {
    throw "Expected property [ColumnNameToAppend] not defined."
}

var lineSeparator = "\n"
var columnSeparator = ","

var flowFile = session.get();

while (flowFile != null) {
    var newColName =
ColumnNameToAppend.evaluateAttributeExpressions(flowFile).getValue()
    var newColValue =
ColumnValueToAppend.evaluateAttributeExpressions(flowFile).getValue()
    flowFile = session.write(flowFile,
        new StreamCallback(function(inputStream, outputStream) {
            var text = IOUtils.toString(inputStream, StandardCharsets.UTF_8)
            var lines = text.split(lineSeparator).filter( function(line) { return
line.length > 0; } )

            outputStream.write((lines.shift() + columnSeparator + newColName +
lineSeparator).getBytes(StandardCharsets.UTF_8))

            outputStream.write(
                lines.map( function(line) {
                    return line + columnSeparator + newColValue;
                }).join(lineSeparator).getBytes(StandardCharsets.UTF_8)
            )
        })
    )
}
```

```

    });
    session.transfer(flowFile, REL_SUCCESS);
    flowFile = session.get()
}

```

## AddOn: Convert Bytes to Hex (Java)

Message digest methods operate on byte array and return byte array. Usually the required output is string, which contains hexadecimal encoding of the byte array.

```

// Convert byte[] to String of hexadecimal
// Method 1: Use Apache Commons codec
import org.apache.commons.codec.binary.Hex;

String foo = "I am a string";
byte[] bytes = foo.getBytes();
System.out.println( Hex.encodeHexString( bytes ) );

```

```

// Convert byte[] to String of hexadecimal
// Method 1: Use DatatypeConverter from Javax XML library
import javax.xml.bind.DatatypeConverter;

String foo = "I am a string";
byte[] bytes = foo.getBytes();
System.out.println( javax.xml.bind.DatatypeConverter.printHexBinary( bytes ) );

```

- 

## Resources

<https://community.hortonworks.com/articles/75032/executescript-cookbook-part-1.html>

<http://openjdk.java.net/projects/nashorn/> - Implements a lightweight high-performance JavaScript runtime in Java with a native JVM. Used in NiFi scripting.

[NiFi Developer's Guide - Processor API](#)