Mule Coding Style Guide

1 Naming Convention

1.1 Application XML Files

Application XML file names are lowercase, hyphen- or underscore-separated.

> Correct: `process-inbound-files.xml`
> Incorrect: `ProcessInboundFiles.xml`

1.2 Flows and Sub-Flows

Flow names are camel-case. When flows in multiple XML files are cross-referenced, name of the flow should be prepended with the name of the XML.

> Correct: `convertCSV2Object,inbound.convertCSV`
> Incorrect: `ProcessCSVFlow, Process-CSV`

1.3 Flow Variables

Flow variables are lower-camel-case.

> Correct: `customerID`
> Incorrect: `OrderID, order-id`

1.4 Properties Files

Properties files are lowercase, must include the file name, environment name and `.properties` extension.

> Correct: `workday.`**`dev`**`.properties, ns-2-sfdc.`**`prod`**`.properties`
> Incorrect: `MyProps, dev.properties`

2   Application XML Source Files

2.1   Encoding

All XML files must be UTF-8 encoded. Always use the following XML declaration:

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

Always use UNIX-style line endings.

2.2   Indentation

XML indentation is 4 whitespaces. Do not use TAB and make sure all tabs are converted to spaces. All nested tags and code within CDATA blocks must use the same indentation.

Example:

```xml
<foreach>
  <expression-component>
    <![CDATA[
        flowVars['AccountClientMap'].put(message.payload.EDI_ClientIDname__c,
          message.payload.Account__c);
            ]]>
        </expression-component>
    </foreach>
```

2.3   Vertical Spacing

Separate semantically unrelated parts of the flow vertically by one blank line.

Example:

```xml
<flow name="contract.GenerateContract">

    <flow-ref name="util.EnrichCustomerData"/>

    <set-variable variableName="homeownerId" value="#[message.payload['SunEdCustId']]"/>
    <set-variable variableName="filename" value="#[message.payload['PricingQuoteId'] +
'_contract.pdf']" />
    <set-variable variableName="folderName"
value="#[message.payload['SunEdCustomer']['PartnerName']]"/>

</flow>
```

Separate flows from each other by single blank line and an optional line of comments.

2.4 Line Wrapping

Break long tags with single new line character; indent second line of attributes so that it is positioned right under the first attribute of the XML element on the line above.

Correct:

```
<x12-edi:config name="GenericEDI" invalidCharacterInValueFail="false" unknownsSegmentFail="false"
                valueLengthErrorFail="false" wrongSegmentsRepeatsFail="false"
                wrongValuesRepeatsFail="false" stringCharacterSet="UNRESTRICTED" doc:name="X12 EDI">
    <x12-edi:schemas>
        <x12-edi:schema>/x12/005010/834.esl</x12-edi:schema>
    </x12-edi:schemas>
</x12-edi:config>
```

Incorrect:

```
<x12-edi:config name="GenericEDI" invalidCharacterInValueFail="false" unknownsSegmentFail="false"
valueLengthErrorFail="false" wrongSegmentsRepeatsFail="false"
wrongValuesRepeatsFail="false" stringCharacterSet="UNRESTRICTED" doc:name="X12 EDI">
    <x12-edi:schemas>
        <x12-edi:schema>/x12/005010/834.esl</x12-edi:schema>
    </x12-edi:schemas>
</x12-edi:config>
```

2.5 Comments

Use `doc:name` and `doc:description` attributes to clearly and concisely document functionality.

See attached MuleESB template projects for examples.

2.6 Structure

2.6.1 Top <mule> Tag

Top <mule> tag contains all namespace and schema location declarations. The namespaces must be declared before schema locations. Each namespace and schema location declaration must start with new line and proper indentation.

Correct:

```
<mule xmlns="http://www.mulesoft.org/schema/mule/core"
      xmlns:sfdc="http://www.mulesoft.org/schema/mule/sfdc"
      xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
      xmlns:spring="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="
http://www.mulesoft.org/schema/mule/sfdc http://www.mulesoft.org/schema/mule/sfdc/current/mule-sfdc.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-current.xsd
http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd">
```

Incorrect:

```
<mule xmlns="http://www.mulesoft.org/schema/mule/core"
xmlns:sfdc="http://www.mulesoft.org/schema/mule/sfdc"
xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
xmlns:spring="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
http://www.mulesoft.org/schema/mule/sfdc http://www.mulesoft.org/schema/mule/sfdc/current/mule-sfdc.xsd http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-current.xsd
http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd">
```

2.6.2 Global Declarations

Global declarations include, but not limited to:
- connector configurations;
- Spring beans;
- properties placeholders;
- processing strategies.

It is recommended to keep all global declarations in a separate XML file (e.g. global.xml or common.xml)

Separate unrelated global declarations vertically by one blank line.

Example:

```
<secure-property-placeholder:config name="Secure_Property_Placeholder" encryptionAlgorithm="Blowfish"
key="${lookup.key}" location="processenrollments.${mule.env}.properties"/>
```

```
        <sfdc:config name="Salesforce__Config" username="${Salesforce_User}" password="${Salesforce_Pwd}"
securityToken="${Salesforce_SecurityToken}" url="${Salesforce_Url}" disableSessionInvalidation="true">
        <reconnect-forever frequency="5000" />
    </sfdc:config>
```

3 Syntax and Best Practices

3.1 MEL Expressions

3.1.1 Flow Variables, Properties, Lists and Maps

Access Map or List element by the key using square brackets [ ].

Correct:

```
flowVars['counter']
message.payload['userId']
message.inboundProperties['params'][0]
```

Not recommended[1]:
```
flowVars.counter
payload.'userId'
message.inboundProperties.'params'(0)
```

3.2 Message Properties

Use `<set-property>` when setting only one message-property.

```
<set-property propertyName="header" value="foobar"/>
```

Use `<message-properties-transformer>` when setting multiple message-properties.

```
<message-properties-transformer>
    <add-message-property key="header" value="foobar"/>
</message-properties-transformer>
```

3.3 Session Properties

Do not use session properties unless there a clear and strong case for doing so; favor flow-variables and message-properties. Flow variables are adequate for passing information between flows as long as one is not going across a "transport barrier"[2].

---

[1] While the dot-notation usually works, there are situations when it does not. It is easier (and advisable) to settle on conventions that *always* work, thus eliminating tedious troubleshooting and mixed styles.
[2] http://blogs.mulesoft.com/dev/anypoint-platform-dev/mule-school-the-mulemessage-property-scopes-and-variables/