Mule Coding Style Guide

1. General

- Unused flows/sub-flows must be removed before committing to version-control
  - Do not comment flows, sub-flows, global-declarations, transformers or components without a dated note explaining why it has not been deleted.
- All flows, sub-flows, components, transformers, scopes (of any kind, including transactions) must be given brief, descriptive names.
  - incorrect: `for each` (uninformative)
  - correct: `for each product` (immediately obvious). Coupled with nearby operations or connectors that refer to a product database or web-service, concise names promote comprehension.
- do not use made-up acronyms - use only industry-standard acronyms unless the customer requires custom acronyms.

WARNING	Do not leave connectors, scopes etc. with default names (eg. Bean, Database, Logger, SMTP, Variable, Choice). The goal is to allow maximum comprehension of flow logic while minimizing the amount of drill-down needed.

2. Naming Convention

2.1. Application XML Files

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

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

2.2. flows, sub-flows, scopes

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

- Correct: processCSVFlow, convertCSV2Object,inbound.convertCSV

- Incorrect: ProcessCSVFlow, Process-CSV

2.2.1. Flow Variables

Flow variables are lower-camel-case.

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

2.3. 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

3. Application Source Files

3.1. Encoding

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

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

Always use UNIX-style line endings.

3.2. Formatting

The guidelines below result in optimum readability; however, most IDEs have different standards, configurability and formatting engines, so it is unlikely that such formatting will survive regeneration, auto-formatting or auto-triggered formatting events.

3.2.1. Indentation

XML indentation is 2 whitespaces. Ensure IDE is configured to convert tabs to spaces. All nested tags and code within CDATA blocks must use the same indentation.

Example

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

3.2.2. Vertical Spacing

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

Example:

```
<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 a single blank line and an optional line of comments.

3.2.3. 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>
```

3.3. Comments

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

See attached MuleESB template projects for examples.

3.4. Structure

3.4.1. Top <mule> Tag

The 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">
```

3.4.2. Global Declarations

Global declarations include, but are not limited to: * connector configs; * spring beans; * properties placeholders; * processing strategies.

All global declarations are to be kept in a dedicated XML file called global.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>
```

3.5. Syntax and Best Practices

3.5.1. MEL Expressions for 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:

```
flowVars.counter
payload.'userId'
message.inboundProperties.'params'(0)
```

3.5.2. Message Properties

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

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

Use <message-properties-transformer> when setting multiple message-properties in one location:

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

3.6. Session Properties

NOTE "Session" in Mule is ***not the same as "session" for web-applications***.

WARNING Do not use session properties! Session properties are serialized and added to message headers when dispatched over some transport boundaries, increasing the size of the message and consuming bandwidth.

- Use flow variables to pass values between flows.
- When crossing transport-boundaries, pass essential data using the header supported by the transport in question, or in the payload