Creating custom templates

This quick tutorial will walk you through the necessary steps to create your own templates for use with the Groovy MDA library.

The groovy MDA library supports three argruments when executing the jar command. Two of them are optional.

```
java -jar groovymda.jar [url to model] <output directory> <url to model processor im
```

In future versions this will mostly likly be changed to support an XML based configuration or maybe use apache commons CLI.

To use your own templates, you can start by creating a Model Processor implementation.

This requires implementing a groovy script that implements a process method.

```
void process(Map context) {

}
```

The recommended approach is to extend the GroovyModelProcessor that comes with the library.

An example working implementation is below:

```
class MyModelProcessor extends GroovyModelProcessor {

        void process(Map context) {
                // ITERATE THROUGH EACH CLASS IN THE MODEL
                getAllClasses(context.model).each { modelElement ->
                        // ADD THE CURRENT MODEL ELEMENT TO THE CONTEXT
                        context.currentModelElement = modelElement

                        // GET THE FULLY QUALIFIED NAME FOR THE CLASS
                        def fullyQualifiedName = getFullyQualifiedName(context.curren

                        // ONLY PROCESS NON JRE CLASSES (java.lang.String does not ne
                        if (!fullyQualifiedName.startsWith("java") && fullyQualifiedN

                                // YOU CAN BIND CLOSURES TO THE CONTEXT TO MAKE THEM 
                                context.helloWorld = { aString ->
                                        return "Hello ${aString}!"
                                }

                                // SET THE TEMPLATE TO USE
                                def templateName = "BasicPojo.gtl"

                                // SET THE OUTPUT FILE NAME FOR THE FULLY QUALIFIED N
                                def outputName = "${fullyQualifiedName.replace('.','/

                                // PROCESS THE TEMPLATE
                                processTemplate(templateName, outputName, context)

                        }
                }
        }

}
```

The above Groovy script will be saved as **MyModelProcessor.groovy**.

The default GroovyModelProcessor script that is in the jar has a lot of helper closures that are bound to the context and available to your templates/processor.

The following are the closures that are bound to the context that are available to use in your templates and model processor:

```
javaToSql(string) - converts camel case to underscore separated
javaType(umlType) - converts a UML data type to java type
firstCharUpper(string) - converts first char to upper case
firstCharLower(string) - converts first char to lower case
getPackageName(modelElement) - gets the package name for the model element
getAttributes(class) - gets the attributes for the class
```

converted by Web2PDFConvert.com

```
getAssociationEnds(model, class) - gets the association ends for the specified class
getEndType(associationEnd) - returns end participant type for single value associatio
getEndName(associationEnd) - get the specified name for end participant or creates one
taggedValues(modelElement) - get tagged values as map for specified model element
isOneToOne(sourceAssociationEnd, targetAssociationEnd) - determine if the source to ta
isOneToMany(sourceAssociationEnd, targetAssociationEnd) - determine if the source to t
isManyToOne(sourceAssociationEnd, targetAssociationEnd) - determine if the source to t
isManyToMany(sourceAssociationEnd, targetAssociationEnd) - determine if the source to
isOwner(association, associationEnd) - determines if the end is the owner in the asso
isCollection(associationEnd) - determine if the association end a collection based on
```

In the above **MyModelProcessor.groovy** example, we are specifying to use a template named **BasicPojo.gtl**

This template will output simple pojos from model elements and also make use of the hello world closure defined in the custom model processor.

The contents of the **BasicPojo.gtl** template are below:

```
<% if (getPackageName(currentModelElement) != "") { %>
package ${getPackageName(currentModelElement)};
<% } %>

/**
 * ${helloWorld(currentModelElement.name)}
 */
public class ${currentModelElement.name} {

        <%
        getAttributes(currentModelElement).each { attribute ->
                def attributeName = attribute.name
                def attributeType = javaType(attribute.type)
                %>
                private ${attributeType} ${attributeName};
                <%
        }
        %>

        <%
        getAssociationEnds(model, currentModelElement).each { sourceEnd ->
                def association = sourceEnd.association
                def targetEnd = association.connection.find { end -> end != sourceEnd
                if (targetEnd.isNavigable()) {
                        def targetType = getEndType(targetEnd)
                        def targetName = getEndName(targetEnd)
                        %>
                private ${targetType} ${targetName};
                        <%
                }
        }
        %>

        <%
        getAttributes(currentModelElement).each { attribute ->
                def attributeName = attribute.name
                def attributeType = javaType(attribute.type)
                %>

                public ${attributeType} get${firstCharUpper(attributeName)}() {
                        return ${attributeName};
                }

                public void set${firstCharUpper(attributeName)}(${attributeType} ${at
                        this.${attributeName} = ${attributeName};
                }

                <%
        }
        %>

        <%
        getAssociationEnds(model, currentModelElement).each { sourceEnd ->
                def association = sourceEnd.association
                def targetEnd = association.connection.find { end -> end != sourceEnd
                if (targetEnd.isNavigable()) {
                        def sourceName = getEndName(sourceEnd)
                        def targetType = getEndType(targetEnd)
                        def targetName = getEndName(targetEnd)
                        %>

                public ${targetType} get${firstCharUpper(targetName)}() {
                        return ${targetName};
                }
```

Web2PDF
converted by Web2PDFConvert.com

```
                public void set${firstCharUpper(targetName)}(${targetType} ${targetNa
                        this.${targetName} = ${targetName};
                }

                        <%

                }
        }
        %>

}
```

Notice the Class documentation section that calls the helloWorld closure that was defined in the **MyModelProcessor.groovy** script.

Any closures bound to the context map passed to the process method of the model processor will be available to the templates.

After adding both files to a directory containing the groovymda.jar and a uml model we can run the following command.

```
java -jar groovymda.jar 'jar:file:./addressbook.zargo!/addressbook.xmi' . file:./MyMo
```

Thats basically it. From this you can create templates to generate any type of code from a uml model.

Example of one of the generated Classes is below:

```
package com.acme.domain;


/**
 * Hello Address!
 */
public class Address {


                private java.lang.String firstName;

                private java.lang.String lastName;

                private java.lang.String street1;

                private java.lang.String street2;

                private java.lang.String city;

                private java.lang.String phone;


                private com.acme.domain.Country country;

                private com.acme.domain.Region region;

                private com.acme.domain.User user;




                public java.lang.String getFirstName() {
                        return firstName;
                }

                public void setFirstName(java.lang.String firstName) {
                        this.firstName = firstName;
                }



                public java.lang.String getLastName() {
                        return lastName;
                }

                public void setLastName(java.lang.String lastName) {
                        this.lastName = lastName;
                }



                public java.lang.String getStreet1() {
                        return street1;
                }
```

converted by Web2PDFConvert.com

```
                  public void setStreet1(java.lang.String street1) {
                          this.street1 = street1;
                  }



                  public java.lang.String getStreet2() {
                          return street2;
                  }

                  public void setStreet2(java.lang.String street2) {
                          this.street2 = street2;
                  }



                  public java.lang.String getCity() {
                          return city;
                  }

                  public void setCity(java.lang.String city) {
                          this.city = city;
                  }



                  public java.lang.String getPhone() {
                          return phone;
                  }

                  public void setPhone(java.lang.String phone) {
                          this.phone = phone;
                  }





                  public com.acme.domain.Country getCountry() {
                          return country;
                  }

                  public void setCountry(com.acme.domain.Country country) {
                          this.country = country;
                  }



                  public com.acme.domain.Region getRegion() {
                          return region;
                  }

                  public void setRegion(com.acme.domain.Region region) {
                          this.region = region;
                  }



                  public com.acme.domain.User getUser() {
                          return user;
                  }

                  public void setUser(com.acme.domain.User user) {
                          this.user = user;
                  }


}
```

© 2007

Web2PDF
converted by Web2PDFConvert.com