

# 1 Instance and Label Generators via patterns

Instance and label generators defined in the generator-policy.xml file using patterns:

## 1.1 SimpleLabel

**Definition in the Generator Policy file:**

```
<generator name="SimpleLabel">  
  <pattern>{label}</pattern>  
</generator>
```

**Usage in the X3ML file:**

```
<label_generator name="SimpleLabel">  
  <arg name="label" type="xpath">text()</arg>  
</label_generator>  
  
<label_generator name="SimpleLabel">  
  <arg name="label" type="constant">The-British-Museum</arg>  
</label_generator>  
  
<label_generator name="SimpleLabel">  
  <arg name="label" type="constant">The-British-Museum</arg>  
  <arg name="language" type="constant">en</arg>  
</label_generator>
```

## 1.2 LocalTermURI

This generator has three arguments, a prefix, a hierarchy and a term. The prefix is a namespace that must be declared in the X3ML definition. The hierarchy and the term can be either constants or variables.

**Definition in the Generator Policy file:**

```
<generator name="LocalThesauriURI" prefix="bmT">  
  <pattern>{hierarchy}/{term}</pattern>  
</generator>
```

**Usage in the X3ML file:**

```
<namespace prefix="bmT" uri="http://collection.britishmuseum.org/id/thesauri/">  
<instance_generator name="LocalTermURI">  
  <arg name="hierarchy" type="constant">TEXT</arg>  
  <arg name="term" type="constant">TEXT</arg>  
</instance_generator>
```

## 1.3 LocalObjectURI

### Definition in the Generator Policy file:

```
<generator name="LocalObjectURI" prefix="bmO">  
    <pattern>{hierarchy}/{object}/{term}</pattern>  
</generator>
```

### Usage in the X3ML file:

```
<namespace prefix="bmO" uri="http://collection.britishmuseum.org/id/" />  
<type>crm:E42_Identifier</type>  
<instance_info>  
    <constant>prn</constant>  
</instance_info>  
<instance_generator name="LocalObjectURI">  
    <arg name="hierarchy" type="constant">object</arg>  
    <arg name="object" type="xpath">text()</arg>  
    <arg name="term" type="constant">prn</arg>  
</instance_generator>  
<label_generator name="SimpleLabel">  
    <arg name="label" type="xpath">text()</arg>  
</label_generator>
```

Similarly other generators can be defined in the generator policy file.

## 2 Custom Instance Generators (via code)

### 2.1 UUID

This function creates a UUID. It does not need to be defined in the generator policy file. It is built in the X3ML engine.

**Usage in the X3ML file:**

```
<instance_generator name="UUID"/>
```

### 2.2 Literal

This function creates a Literal. It does not need to be defined in the generator policy file. It is built in the X3ML engine.

**Usage in the X3ML file:**

```
<instance_generator name="Literal">
  <arg name="text" type="xpath">text()/</arg>
  <arg name="language" type="constant">de</arg>
</instance_generator>
```

### 2.3 URIorUUID

This function checks the argument “text” and if it is a valid URI it copies it to the rdf output otherwise it creates a UUID.

**Definition in the Generator Policy file:**

```
<generator name="URIorUUID">
  <custom generatorClass="gr.forth.URIorUUID">
    <set-arg name="text"/>
  </custom>
</generator>
```

**Usage in the X3ML file:**

```
<instance_generator name="URIorUUID">
  <arg name="text" type="xpath">text()/</arg>
</instance_generator>
```

### 2.4 Dates

Currently the two functions GermanDateTime and BMdates are the same.

**Definition in the Generator Policy file:**

```
<generator name="GermanDateTime">
  <custom generatorClass="gr.forth.GermanDate">
```

```
        <set-arg name="bound" type="constant"/>
        <set-arg name="text" type="xpath"/>
    </custom>
</generator>
```

**Definition in the Generator Policy file:**

```
<generator name="BMdates">
    <custom generatorClass="gr.forth.GermanDate">
        <set-arg name="bound" type="constant"/>
        <set-arg name="text" type="xpath"/>
    </custom>
</generator>
```

**Usage in the X3ML file:**

```
<instance_generator name="BMdates">
    <arg name="bound">Upper</arg>
    <arg name="text">text()</arg>
</instance_generator>

<instance_generator name="GermanDateTime">
    <arg name="bound">Lower</arg>
    <arg name="text">text()</arg>
```

## 3 Custom Label Generators (via code)

### 3.1 Literal

This function creates an **rdfs: label** if used as a label generator. It does not need to be defined in the generator policy file. It is built in the X3ML engine.

**Usage in the X3ML file:**

```
<label_generator name="Literal">
  <arg name="text" type="xpath">text()</arg>
  <arg name="language" type="constant">de</arg>
</label_generator>
```

### 3.2 prefLabel

This function creates a **skos:prefLabel**. It does not need to be defined in the generator policy file. It is built in the X3ML engine.

**Usage in the X3ML file:**

```
<label_generator name="prefLabel">
  <arg name="text" type="xpath">text()</arg>
  <arg name="language" type="constant">de</arg>
</label_generator>
```

### 3.3 Constant

This function creates an **rdfs: label** as Literal but it is always a constant. It does not need to be defined in the generator policy file. It is built in the X3ML engine.

**Usage in the X3ML file:**

```
<label_generator name="Constant">
  <arg name="text" type="constant">a constant value</arg>
  <arg name="language" type="constant">en</arg>
</label_generator>
```

**It is equivalent to:**

```
<label_generator name="Literal">
  <arg name="text" type="constant">a constant value</arg>
  <arg name="language" type="constant">en</arg>
</label_generator>
```