

# ProjectX de Sun

*Michel CASABIANCA*  
*[casa@sweetohm.net](mailto:casa@sweetohm.net)*

Le parser XML de Sun (nom de code "ProjectX") présente un défaut de localisation. Cet article se propose d'y remédier.

## Un bug de localisation

Le [parser XML de Sun](#) est réputé pour son respect des normes et la clarté de ses messages d'erreur (voir le test de [Developer Life](#)). Il est donc tout indiqué pour programmer un validateur de fichiers XML.

Le code suivant est un exemple d'un tel validateur (simple pour la démonstration) :

```
import org.xml.sax.*;
import org.xml.sax.helpers.*;

public class Validator
{
    /** parser XML validant de ProjectX */
    private static final String parserClassName=
        "com.sun.xml.parser.ValidatingParser";

    /**
     * Méthode main
     *
     * @param args le fichier à valider
     */
    public static void main(String[] args) {
        try {
            // on instancie le parser
            Parser parser=ParserFactory.makeParser(parserClassName);
            // on construit la source du fichier à valider
            String fichier="file:"+
                System.getProperty("user.dir")+
                System.getProperty("file.separator")+
                args[0];
            InputSource source=new InputSource(fichier);
            // on parse le fichier
            parser.parse(source);
        }
        // on intercepte les exceptions de type SAXParseException
        // qui encapsulent les messages d'erreur XML
        catch(SAXParseException e1) {
            System.out.println(e1.getSystemId()+" "+
                e1.getLineNumber()+" "+
                e1.getMessage());
        }
        // on affiche les autres exceptions
    }
}
```

```
        catch(Exception e2) {e2.printStackTrace();}
    }
}
```

Ce programme est très simple et je n'expliquerai pas son fonctionnement dans cet article (la programmation d'applications SAX pourrait bien faire prochainement l'objet d'un article).

Pour lancer le programme, il faut bien sûr inclure l'archive *xml.jar* du parser du ProjectX dans le CLASSPATH. Pour lancer ce programme, on pourra utiliser (sous UNIX) le script suivant :

```
#!/bin/bash
# on désactive le compilateur JIT
export JAVA_COMPILER=
# répertoire du validateur (à modifier)
VALID_HOME=/home/casa/tmp/test
# définition du CLASSPATH
export CLASSPATH=$VALID_HOME/:$VALID_HOME/xml.jar
# lancement du validateur
java Validator $*
```

Essayons maintenant notre validateur sur un simple fichier d'exemple, comme celui-ci :

```
<?xml version="1.0"?>

<exemple>
<exemple>
```

On obtient alors le message d'erreur suivant :

```
file:/home/casa/tmp/test/test.xml:4:com.sun.xml.parser/P-035
```

Le moins que l'on puisse dire, c'est que ce n'est pas clair ! Le problème vient du fait que la locale française n'existe pas pour les messages d'erreur. On pourrait cependant s'attendre à ce que le parser passe à la locale anglaise, mais ce n'est pas le cas...

## Première solution : changer la locale

La première solution consiste à demander au parser de changer de locale pour la locale *en\_US*, avec la ligne suivante :

```
parser.setLocale(new java.util.Locale("en", "US"));
```

Le code de notre validateur de fichiers devient alors :

```
import org.xml.sax.*;
```

```

import org.xml.sax.helpers.*;

public class Validator
{
    /** parser XML validant de ProjectX */
    private static final String parserClassName=
        "com.sun.xml.parser.ValidatingParser";

    /**
     * Méthode main
     *
     * @param args le fichier à valider
     */
    public static void main(String[] args) {
        try {
            // on instancie le parser
            Parser parser=ParserFactory.makeParser(parserClassName);
            // on change la locale
            parser.setLocale(new java.util.Locale("en", "US"));
            // on construit la source du fichier à valider
            String fichier="file:"+
                System.getProperty("user.dir")+
                System.getProperty("file.separator")+
                args[0];
            InputSource source=new InputSource(fichier);
            // on parse le fichier
            parser.parse(source);
        }
        // on intercepte les exceptions de type SAXParseException
        // qui encapsulent les messages d'erreur XML
        catch(SAXParseException e1) {
            System.out.println(e1.getSystemId()+":"+
                e1.getLineNumber()+":"+
                e1.getMessage());
        }
        // on affiche les autres exceptions
        catch(Exception e2) {e2.printStackTrace();}
    }
}

```

Si l'on parse notre document, on obtient maintenant le message suivant :

```

file:/home/casa/tmp/test/test.xml:4:End of entity not allowed;
an end tag is missing.

```

Ce qui est tout de même mieux !

## Deuxième solution : correction des sources

Une deuxième solution consiste tout simplement à corriger ce bug, de manière à ce que le parser adopte la locale *en\_US* si la locale de la VM n'est pas supportée.

Pour ce faire, il nous faut modifier la méthode `init()` du fichier `com/sun/xml/parser/Parser` des sources comme suit :

Remplacer :

```
if (locale == null)
    locale = Locale.getDefault ();
```

Par :

```
if (locale == null) {
    locale = Locale.getDefault ();
    // casa: test if default locale is supported
    if(!messages.isLocaleSupported (locale.toString ()))
        locale = new Locale("en", "US");
}
```

Ainsi, on vérifie si la locale par défaut de la VM est supportée et on passe à la locale `en_US` si ce n'est pas le cas.

Pour recompiler les sources, on pourra utiliser le makefile suivant (à placer dans le répertoire des sources) :

```
all: class jar
class:
    javac -O `find -name "*.java"`
jar:
    jar cf xml.jar `find -name "*.class"` `find -name "*.properties"`
clean:
    rm -f `find -name "*.class"`
```

On notera que cette recompilation est la bienvenue car le fichier `jar` fourni par Sun a été compilé avec l'option de débogage activée (option `-g` de la ligne de commande). La recompilation des sources permet ainsi de gagner en vitesse d'exécution (le gain est cependant minime) et en place.

## Troisième solution : traduction des messages

La troisième solution est certainement la meilleure : la traduction des messages d'erreur. Pour ce faire, il faut fournir au parser des fichiers `Messages_fr.properties` à placer dans les répertoires `com.sun.xml.parser.resources` (pour les messages du parser SAX) et `com/sun/xml/tree/resources/Messages_fr.properties` (pour les messages du parser DOM) des sources. Il faut ensuite recompiler le tout (avec le makefile ci-dessus). Les messages d'erreur sont alors affichés en français.

J'ai traduit les messages d'erreur pour les parsers SAX et DOM. Suivre [ce lien](#) pour les télécharger. Cette archive contient également les fichiers d'exemple de cet article.