

L'internationalisation

L'INTERNATIONALISATION	1
CONTENU DE LA SECTION	2
QUELQUES GENERALITES	3
<i>les besoins</i>	3
<i>les moyens</i>	3
L'INTERNATIONALISATION DU PLUGIN.XML	4
<i>l'externalisation des strings dans le plugin.xml</i>	4
L'INTERNATIONALISATION DU CODE JAVA	5
<i>l'externalisation des strings dans le code java</i>	5
L'INTERNATIONALISATION DU CODE JAVA (2)	6
<i>l'externalisation des strings dans le code java (suite)</i>	6

Quelques généralités

les besoins

- réaliser des applications multilingues : adaptation de l'UI (traduction des informations textuelles)
 - Labels des widgets dans les fenêtres
 - messages d'erreurs, alertes, dialogues, ...
 - documentation
 - About
- Prise en compte des variables d'environnement pour personnaliser l'UI

les moyens

- externaliser les Strings intervenant dans les plugins dans des fichiers .properties
 - plugin.xml → plugin.properties
 - plugin_<language>_<country>.properties : plugin_de.properties, plugin.jp.properties
- ISO 639
ISO 3166
- externaliser les Strings intervenant dans les classes dans des fichiers <package>_<language>_<country>.properties
 - les Strings à adapter vs les Strings invariants (IDs utilisés en interne à l'infrastructure)
 - mécanisme des Fragments (voir suite)

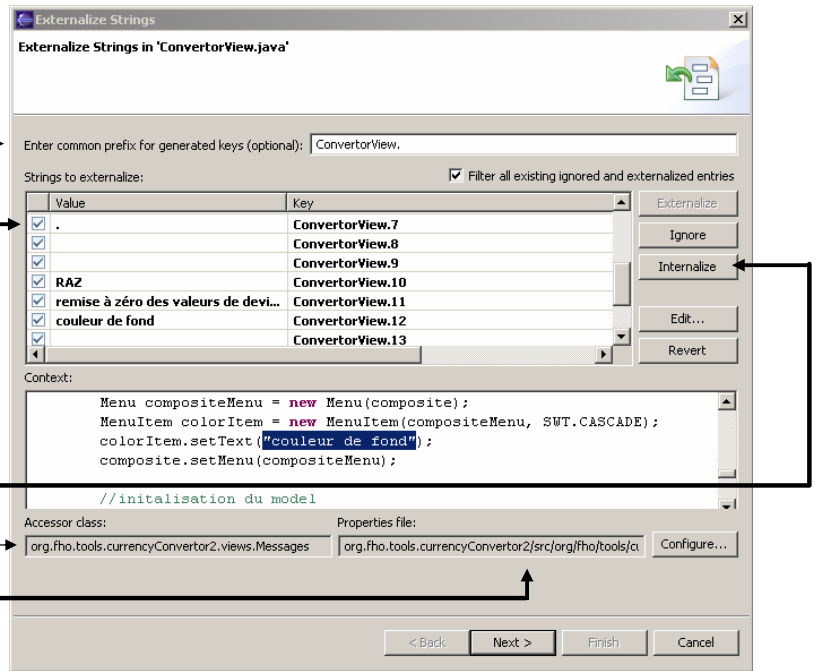
l'externalisation des strings dans le code java

- utilisation de la commande :
Format > externalize Strings

sélection des strings à externaliser
définition d'un préfixe
Externalize vs Ignore vs Internalize

Classe d'accès :
offre la méthode static getString(String aKey)

nom du fichier .properties contenant les Strings



l'internationalisation du code java (2)

l'externalisation des strings dans le code java (suite)

- création d'une classe d'accès par l'utilitaire de refactoring
- création d'un fichier .properties par l'utilitaire de refactoring

```
package org.fho.tools.currencyConvertor2.views;

import java.util.MissingResourceException;

/**
 * @author Horn
 */
public class Messages {
    private static final String BUNDLE_NAME = "org.fho.tools.currencyConvertor2.views.messages"; // $NON-NLS-1$

    private static final ResourceBundle RESOURCE_BUNDLE = ResourceBundle.getBundle(BUNDLE_NAME);

    private Messages() {}

    public static String getString(String key) { // TODO Auto-generated method stub
        try {
            return RESOURCE_BUNDLE.getString(key);
        } catch (MissingResourceException e) {
            return '!' + key + '!';
        }
    }
}

ConvertorView.0=modifyText exception :
ConvertorView.10=RAZ
ConvertorView.11=remise \u00E0 z\u00E9ro des valeurs de devise
ConvertorView.12=couleur de fond
```

classe d'accès

Non Locale String