

# GNU gettext pour .NET

## Internationalisation et régionalisation

Vue d'ensemble de solution



Site Gettext.NET

<http://sourceforge.net/projects/gettextnet/>

Serguei TARASSOV  
Juillet 2012

# GNU gettext

- Qu'est que i18n et l10n ?
- Qu'est que GNU gettext ?
- Le procédé GNU
- Formes plurielles et contexte de traduction
- Rôles et points de vues dans le procédé
  - Rôle « Programmeur »
  - Rôle « Intégrateur »
  - Rôle « Traducteur »
  - Point de vue « Utilisateur »

# Internationalization (i18n)

- L'internationalisation d'un logiciel consiste à **préparer son adaptation** à des langues et des cultures différentes
- Un **travail essentiellement technique**, mené par des programmeurs
- GNU gettext

- Code non-adapté

```
Console.WriteLine("Hello, world!");
```

- Code adapté

```
Console.WriteLine(catalog.GetString("Hello, world!"));
```

# Localisation (I10n)

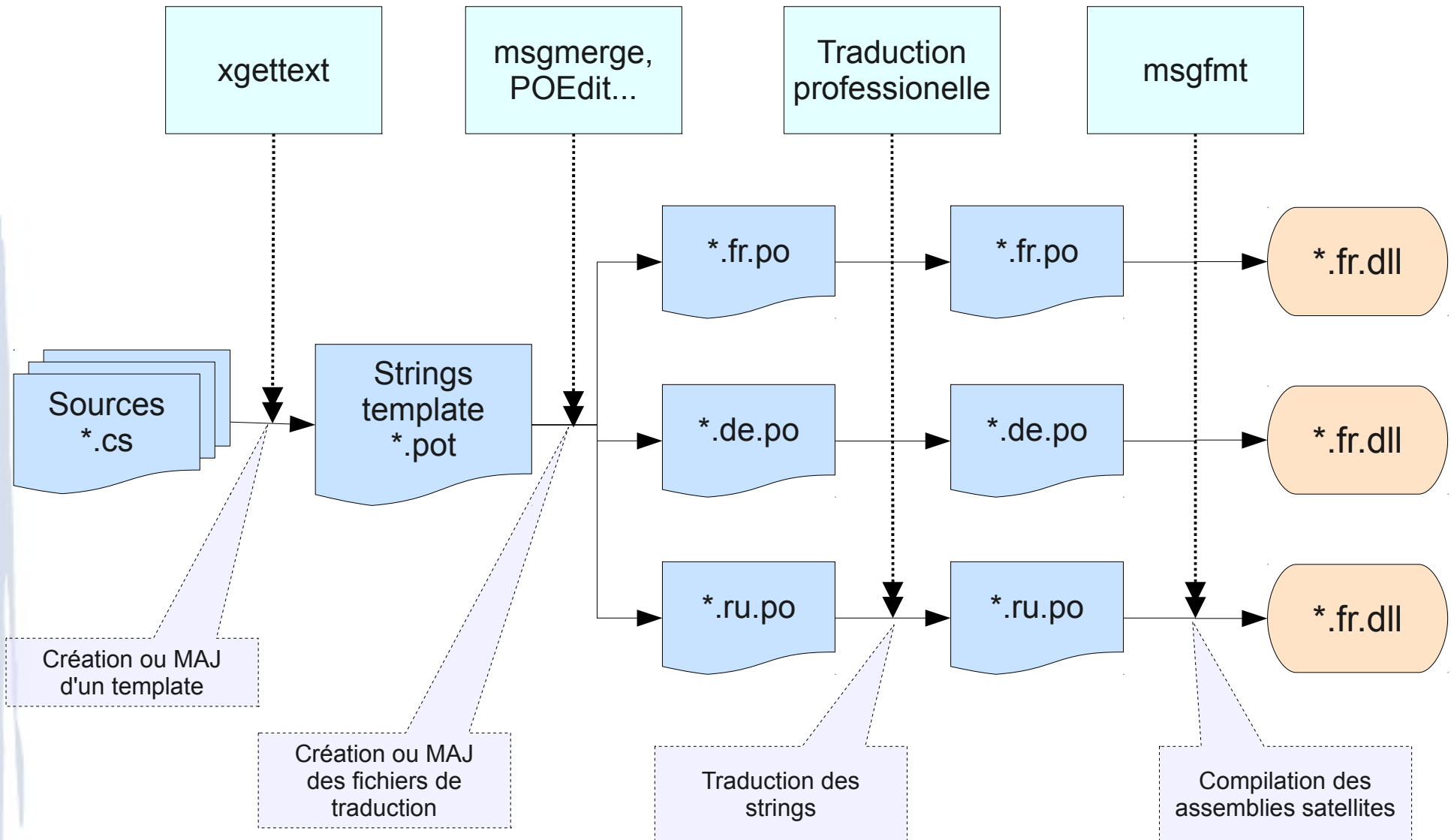
- La régionalisation de logiciel concerne le **processus de traduction** de l'interface utilisateur d'un logiciel d'une langue vers une autre et en l'adaptant à la culture locale
- Un **travail essentiellement linguistique et culturel**, mené par des traducteurs et/ou des experts fonctionnels
- GNU gettext

```
#: src/name.cs:36  
msgid "My name is {0}"  
msgstr "Je m'appelle {0}"
```

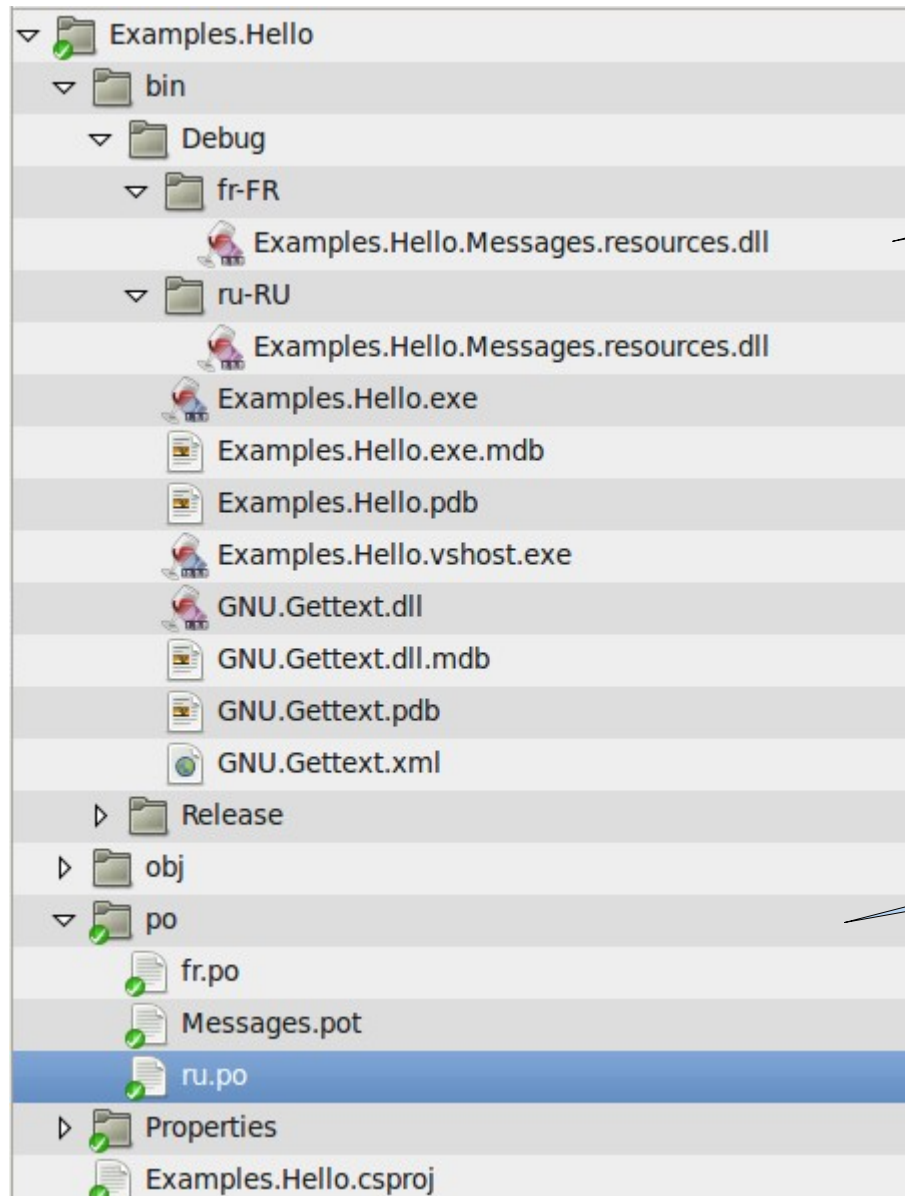
# GNU gettext

- Ensemble des standards et des technologies pour i18n
- Implémentations multiplateformes
  - OS : Linux/Solaris, Windows, Mac
  - Plateformes : Java, .NET
  - Langages et environnements : C/C++, Qt, Shell, Python, Ruby, Visual Basic, Delphi/Pascal etc.
- Plusieurs outils pour l10n
- Gratuit et opensource (GPL et LGPL licences)
- Premier standard publié en 1988

# Le procédé GNU (.NET)



# Structure du projet



Assemblies  
satellites

Fichiers \*.pot et \*.po



# Formes plurielles

- Traduction est choisi en fonction de nombre N

- Origine en Français (2 formes)

- *J'ai 1 pomme*
    - *J'ai 2 pommes*

- Code C#

```
GetPluralString("J'ai {0} pomme", "J'ai {0} pommes", n);
```

- Résultat en Russe (3 formes)

- *У меня 1 яблоко*
    - *У меня 2 яблока*
    - *У меня 10 яблок*



# Formes plurielles

- Exemples pour toutes les langues principales :
  - <http://translate.sourceforge.net/wiki/l10n/pluralforms>
- DE
  - `nplurals=2; plural=(n != 1)`
- FR
  - `nplurals=2; plural=(n > 1)`
- RU
  - `nplurals=3; plural=(n%10==1 && n%100!=11 ? 0 : n%10>=2 && n%10<=4 && (n%100<10 || n%100>=20) ? 1 : 2)`

# Contexte de traduction

- Traduction est choisi en fonction du contexte
- Origine en Anglais (1 phrase)

- *Text encoding*

- Code C#

```
GetParticularString("Computers", "Text encoding");
```

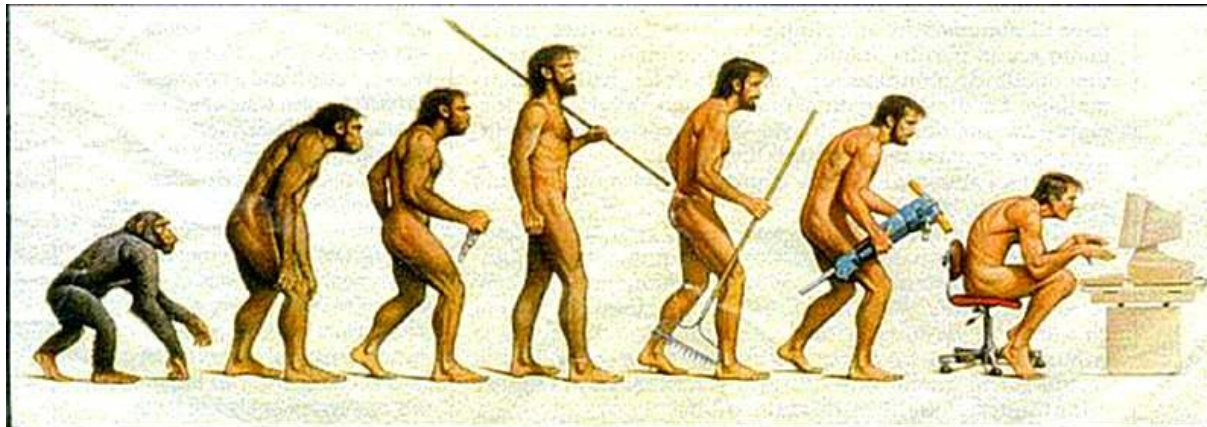
```
GetParticularString("Military", "Text encoding");
```

- Résultat en Francais (2 phrases)

- *Encodage du texte*

- *Chiffrement du texte*

# Rôles et points de vue des collaborateurs dans le procédé



# Rôle « Programmeur »



- Créer le catalogue

```
GettextResourceManager catalog =  
    new GettextResourceManager("Examples.Hello.Messages");
```

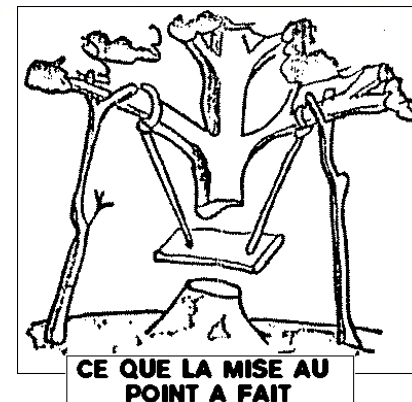
- Entourer les strings dans le code

```
catalog.GetString("Hello, world!");  
catalog.GetStringFmt(  
    "This program is running as process number \"{0}\".",  
    Process.GetCurrentProcess().Id);
```

- Ne fait rien pour le code de design (i.e. WinForms) mais appeler la méthode `GNU.Gettext.WinForms.Localizer.Localize(form1, catalog);`

```
GNU.Gettext.WinForms.Localizer.Localize(form1, catalog);
```

# Rôle « Intégrateur »



- MAJ le template \*.pot

```
GNU.Gettext.Xgettext.exe -dD:\App1\Sources -r -oD:\App1\po\Messages.pot
```

- Optionnellement, MAJ les fichiers de traductions \*.<culture>.po (cf traducteur)

```
msgmerge D:\App1\po\Messages.fr.po D:\App1\po\Messages.pot
```

- Contrôler le nombre ou % des éléments non-traduits

```
msgattrib --untranslated D:\App1\po\Messages.fr.po
```

- Compiler les assemblées satellites

```
GNU.Gettext.Msgfmt.exe -iD:\App1\po\Messages.fr.po -lfr-FR -dD:\App1\Bin\Debug  
-bApp1.Messages
```

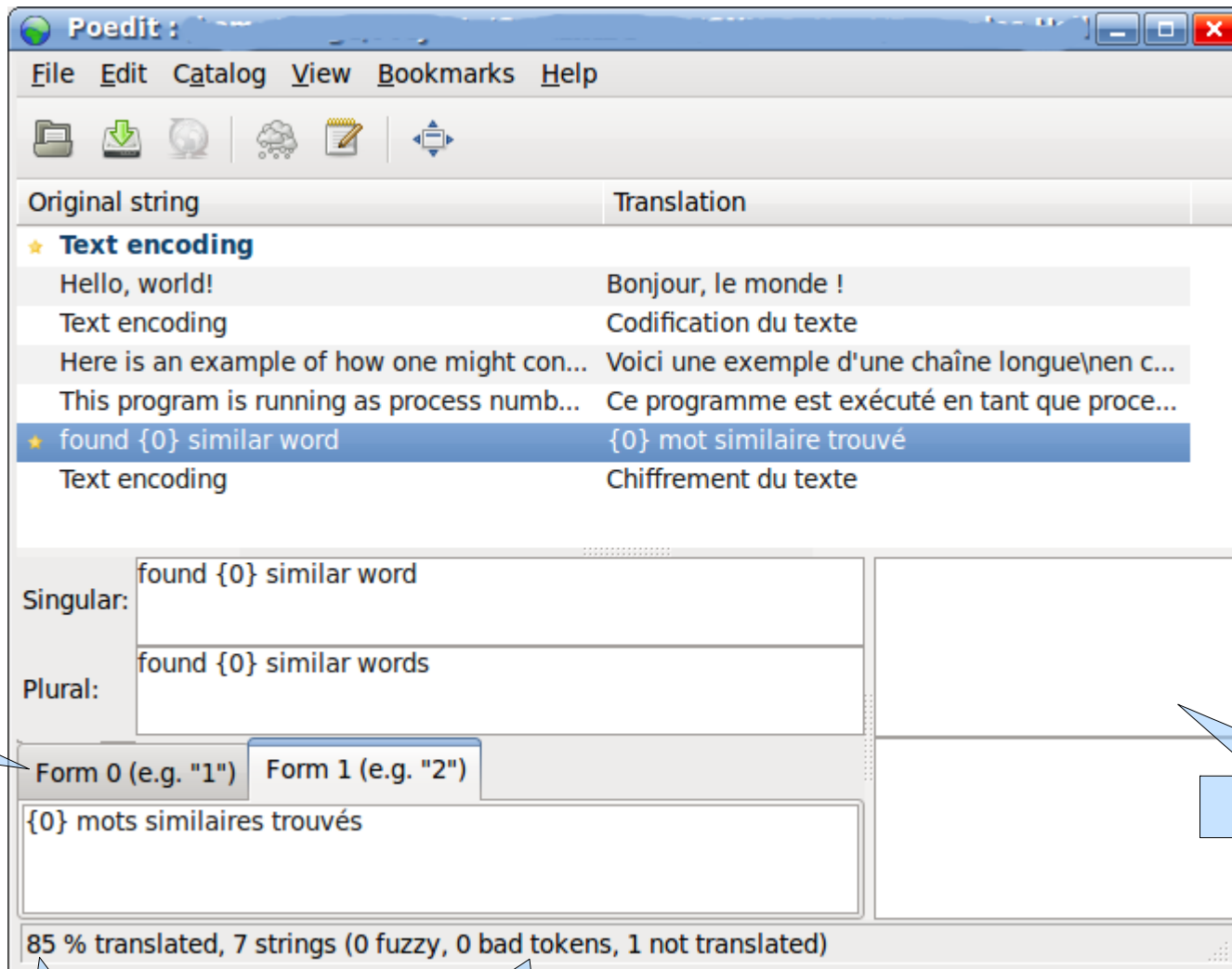
# Rôle « Traducteur »

- Choisir un éditeur PO (i.e POEdit)
- Récupérer un fichier \*.po
- Établir la méta-information (auteur, forme pluriel, langue etc.)
- Optionnellement, MAJ ce fichier depuis le template \*.pot
- Traduire les strings
- Archiver le fichier





# POEdit (native Windows/Linux/Mac)



Formes  
plurielles

% de traduction

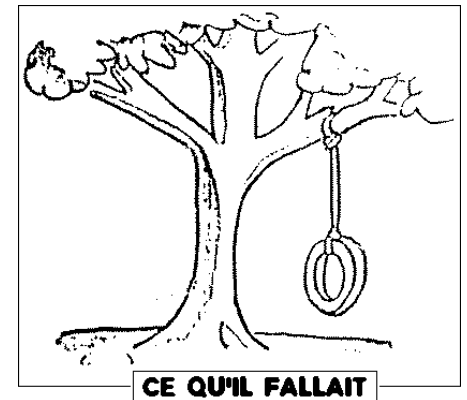
Formatage incorrect

Commentaires



# Point de vue utilisateur

- Changer la région au niveau d'application
- Changer la région au niveau d'environnement (i.e. Windows) pour l'ensemble des applications



# Points clés

- Méthode décentralisée
- Procédé complet
- Supporte multiplateforme, multiple-environnement et langage indépendant (i.e. méthode MS n'existe que dans Visual Studio)
- Standards approuvés et ouverts depuis 1988
- Plusieurs outils/implémentations disponibles

Merci !  
Questions ?