

■ Le site sur les technologies .Net



dotnet-tech.com

■ Actualité ■ Tutoriels ■ Liens ■ Contact

Localisez vos applications ASP.NET (VB.NET uniquement)

Nouveau ! Téléchargez le Code source exemple gratuit [VB.NET ici](#) et [C# là](#)

Mis à jour le 19/03/2003

Par [Elise Dupont](#)

Droit de diffusion:

L'ensemble ou partie de ce document ainsi que le code mis à disposition, ne peut être diffusé sur d'autres sites Web sans l'autorisation au préalable de son créateur.

Avant Propos :

Ce document a pour but d'aider les développeurs qui connaissent déjà un peu l'environnement ASP.NET à gérer de la façon la plus simple possible plusieurs langues, ce qui est souvent le cas quand on développe un site Web qui se veut le plus ouvert possible.

Les termes comme " WebForm " ne seront pas expliqués car s'il fallait tout reprendre du début ce tutorial serait un roman :) Pour les débutants, je vous conseille donc de vous faire la main sur de simples applications asp.net avec une seule langue.

De plus la solution que je propose ici a été testée et validée, mais il existe peut être d'autres solutions différentes pour parvenir au même but. La solution proposée ici suit tout simplement les recommandations données par Microsoft sur le site MSDN : <http://www.microsoft.com/france/msdn/default.asp>

* **Localisation** : Adaptation, principalement du logiciel, sur les plans linguistique, culturel, ergonomique, religieux, etc., pour un groupe cible qui appartient à une zone géographique précise.

Sommaire:

- 2. Convention sur les noms des langues
- 3. Quelle langue est chargée par l'application Web
- 4. Organisez vos fichiers ressource
 - 4.1 Les répertoires
 - 4.2 Fichiers .resx
- 5. Le code
 - 5.1 Le fichier resx
 - 5.2 Le Global.asax
 - 5.3 Le WebForm
- 6. Compiler vos ressources dans une DLL sans compiler tout le projet
 - 6.1 RESGEN.exe
 - 6.2 AL.exe
 - 6.3 Mettre les fichiers au bon endroit
- 7. SOS ça marche pas !!!
- 8. La doc qui peut vous sauver la vie
- 9. Conventions de noms, la liste

INFO DE DERNIERE MINUTE : Il semblerait, suite aux retours et conseils de Pascal Lebouc, que la solution exposée dans ce cours ne fonctionne que pour VB.NET! Ce qui est surprenant car contredirait le fait que sous .NET "tout" est identique quel que soit le langage. Avant d'en savoir plus sur le sujet, je vous conseille donc de ne pas accorder trop d'importance à cet article si vous développez en C# (j'ai d'ailleurs supprimé le code C# pour éviter toute confusion). Je m'empresserais de mettre le cours à jour quand j'aurais plus d'informations. A la place, je vous conseille plutôt de télécharger les Codes source exemple gratuits [VB.NET ici](#) et [C# là](#), en fonction de votre langage de prédilection, et de vous en inspirer pour vos projets. Je vous remercie de votre compréhension et de votre patience, je ferais de mon mieux pour vous tenir informé.

1. Système de fonctionnement :

Le but est de séparer le texte affiché de l'application afin que l'application soit indépendante des langues, et que donc quand vous modifiez le texte, l'application ne soit pas recompilée à chaque fois, ce qui devient important quand on manipule plusieurs langues. Les principes de base sont :

- Le texte à afficher (nom d'un label, d'un textbox, d'un bouton radio, titre d'une page etc.) est stocké dans des fichiers spécifiques. On appelle ces fichiers des fichiers ressource (extension : .resx).
- Les fichiers ressources (fichier .resx) sont séparés et triés dans un dossier de votre application qui ne contiendra que des fichiers de localisation.
La façon dont vous trie les fichiers est très importante et très stricte comme vous le verrez plus tard. Quant au contenu de ces fichiers il est très simple : à gauche l'identifiant à droite la chaîne de caractères correspondante.
- L'application Web détectera la langue qui a été sélectionnée dans les paramètres du navigateur et chargera la langue correspondante. Ainsi si vous avez deux navigateurs avec chacun une langue différente, chaque navigateur aura un contenu différent.
- Il faut définir une langue par défaut. Il n'y a que cette langue qui sera compilée avec le projet à chaque modification. La langue par défaut est à part de toutes les autres.
- Il y a seulement UN SEUL fichier .resx (dit aussi fichier ressource) par

WebForm.

Nouveau ! Téléchargez le source VB.NET d'exemple disponible ici et la version C# disponible là pour vous aider à suivre les explications en temps réel.

2. Convention sur les noms des langues :

Vous ne pouvez pas utiliser les noms que vous voulez pour les langues que vous ajoutez à votre application web. Vous devez suivre les spécifications données par la RFC 1766 (<http://www.ietf.org/rfc/rfc1766.txt>) Une liste complète de toutes les langues et les codes correspondants peut être trouvée à la fin de cette documentation.

3. Quelle langue est chargée par l'application Web :

.NET a tout un système pour charger automatiquement la bonne langue. Ce système s'appelle le "Fallback process ", que nous allons appeler FP pour plus de facilité. Si on demande une langue qui n'a pas été définie dans le projet Web, le FP va tenter de trouver la langue qui s'en approche le plus, et ne générera une erreur qu'en dernier ressort. Le FP fonctionne selon les étapes suivantes :

- Le FP cherche en premier la langue qui a été spécifiée.
- S'il ne la trouve pas dans le projet, il cherche au niveau supérieur (exemple : s'il ne trouve pas français canadien, il cherche s'il y a un français tout court)
- S'il ne trouve rien il charge la langue par défaut
- Si la langue par défaut n'a pas été précisée ou incluse dans le projet, alors le FP propage une erreur de type "Cannot find specified resource"

Vous voyez ainsi donc l'intérêt de définir une langue par défaut : c'est le dernier recours si aucune langue n'a été trouvée. Cela évite donc d'obtenir une erreur et de toujours avoir une langue de dernier secours.

Exemple :

Supposons que le navigateur client soit configuré avec la langue "Espagnol Mexicain ".

- Si l'on a bien suivi les conventions pour nommer les langues, le FP va en premier chercher la ressource "es-MX".
- S'il ne la trouve pas, il cherche au niveau supérieur (c'est-à-dire simplement Espagnol "es").
- Si on n'a pas défini de ressource pour l'Espagnol le FP cherche tout autre niveau parent qui pourrait potentiellement correspondre à l'Espagnol Mexicain et dont le fichier ressource existe.
- Si par défaut vous avez mit la langue " en " (Anglais) et que le FP ne trouve rien pour l'Espagnol Mexicain, il charge l'Anglais.

- Si vous n'avez aucune langue par défaut, vous aurez une erreur.

4. Organisez vos fichiers ressource :

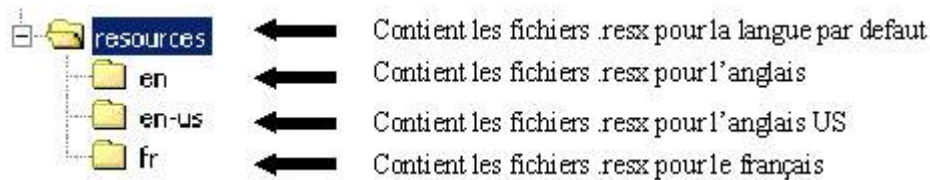
Voici l'organisation recommandée. Une autre organisation pourrait ne pas fonctionner.

Imaginons que votre application s'appelle MyApp. Et le chemin de votre application est : C:\Inetpub\wwwroot\MyApp.

4.1 Les répertoires

Dans l'explorateur de projets, ajoutez un répertoire "Resources" dans votre application web.

Puis les fichiers doivent être placés de la façon suivante :



Comme vous pouvez le voir, il faut un répertoire par langue. Tous ces répertoires sont placés en dessous du répertoire "Resources ". Vous ne devez pas regrouper par exemple les langues " anglais US " et " anglais canadien " sous le groupe " anglais ", même si en fait " anglais US " et " anglais canadien " sont un sous niveau de "anglais ". Tous les répertoires doivent être au même niveau.

4.2 Fichiers .resx

La façon de nommer les fichiers .resx est stricte. Si vous ne suivez pas les recommandations suivantes cela pourrait ne pas fonctionner.

Le fichier resx de la langue par défaut doit être nommé ainsi :

Nom_du_webform.resx

Les autres resx ainsi:

nom_du_webform.code_langue.resx

Dans notre exemple nous avons deux WebForms nommés WebForm_A et WebForm_B. Ce qui nous donne donc :

```
Resources:
  WebForm_A.resx
  WebForm_B.resx
en:
  WebForm_A.en.resx
  WebForm_B.en.resx
en-us:
```

```
WebForm_A.en-us.resx
WebForm_B.en-us.resx
fr:
WebForm_A.fr.resx
WebForm_B.fr.resx
```

Dernier point important :

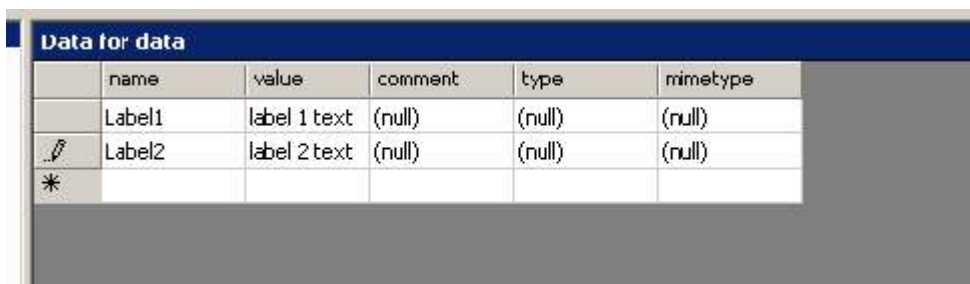
N'oubliez pas de supprimer tous les fichiers resx qui sont générés automatiquement par .NET pour chaque WebForm, fichiers vides donc sans intérêt. Ces fichiers sont parfois cachés. Pour les voir cliquez sur "Afficher tous les fichiers" et regardez sous le WebForm, le fichier resx généré par .NET est là. Si vous ne supprimez pas ces fichiers vous aurez une erreur à la compilation.

5. Le code :

Voici "co" implémenter la détection automatique des langues dans votre application.

5.1 Le fichier resx

Prenons l'exemple d'un WebForm qui a deux labels. Dans votre fichier ressource vous mettez le contenu à afficher pour vos labels :



	name	value	comment	type	mimetype
	Label1	label 1 text	(null)	(null)	(null)
	Label2	label 2 text	(null)	(null)	(null)
*					

5.2 Le Global.asax

Visual Basic .NET

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    'ResourceManager pour WebForm1
    Application("RM4WebForm1") = New System.Resources.ResourceManager(GetType(WebForm1))
    'ResourceManager pour WebForm2
    Application("RM4WebForm2") = New System.Resources.ResourceManager(GetType(WebForm2))
End Sub

Sub Application_BeginRequest(ByVal sender As Object, ByVal e As EventArgs)
    ' Se lance à chaque début de requête du client
    ' Pour chaque requête, initialiser la culture de l'application avec la langue du navigateur
    Try
        Thread.CurrentThread.CurrentCulture = CultureInfo.CreateSpecificCulture(Request.UserLanguages(0))
    End Try
End Sub
```

```

Catch ex As Exception
    ' Si erreur mettre "fr" par exemple
    Thread.CurrentThread.CurrentCulture = New CultureInfo("fr")
End Try
Thread.CurrentThread.CurrentUICulture =
Thread.CurrentThread.CurrentCulture
End Sub

```

Explications :

Request.UserLanguages(0) : Récupère la langue définie dans le navigateur.

Thread.CurrentThread.CurrentCulture : Définit la culture pour la session courante.

5.3 Le WebForm

Visual Basic .NET

```

Private Sub Page_Init(ByVal sender As Object, ByVal e As
EventArgs) Handles MyBase.Init ' Récuperer ResourceManager
de l'Application
    rm = Application("RM4WebForm1")
End Sub

Private Sub Page_Load(ByVal sender As Object, ByVal e As
EventArgs) Handles MyBase.Load ' Récuperer ResourceManager
de l'Application
    InitializeComponent()
    Label1.Text = rm.GetString("Label1")
    Label2.Text = rm.GetString("Label2")
End Sub

```

Explications :

Vous récupérez le contenu d'un fichier ressource avec la méthode **GetString("ma_clef")**.

Par ailleurs, pour le code **Label1.Text** cela ne peut fonctionner avec un contrôle HTML, à moins que celui-ci soit explicitement converti en contrôle serveur (avec `Convert.FromHtml` événements).

Si vous voulez appliquer cet exemple à un contrôle HTML vous devrez plutôt utiliser **Label1.InnerText = rm.GetString("Label1")**. (merci à [MonPierrot](#) pour la précision).

6. Compiler vos ressources dans une DLL sans compiler tout le projet :

Le but de cette documentation était en partie de vous permettre de pouvoir effectuer autant de changements que vous désirez dans les fichiers ressources sans avoir à toucher au projet en lui-même. Cette partie est la plus complexe, mais aussi la plus intéressante. Si les dll ne sont pas générées correctement, rien ne fonctionnera, et la langue par défaut sera toujours chargée à la place des autres.

Deux outils vont vous aider dans cette tâche :

resgen.exe (Resources generator) et al.exe (Assembly linker)

Premièrement, vérifiez bien que vous avez nommé vos fichiers ressources EXACTEMENT comme cela a été décrit plus haut. Ensuite, la génération de la dll est en deux étapes :

- Transformer votre fichier .resx en un fichier .resources
- Lier tous les fichiers .resources (un fichier par WebForm) dans une même dll (une dll par langue)

Pour ces deux outils, utilisez l'invite de commande fourni par Visual Studio (Visual Studio .NET Command Prompt).

6.1 RESGEN.exe

Transformez vos fichiers .resx en fichiers .resources ainsi:

```
>resgen.exe [le fichiers resx]
```

Par

exemple:

Pour le fichier .resx nommé WebForm_A.en.resx la commande sera:

```
>resgen WebForm_A.en.resx
```

Et le fichier généré sera : WebForm_A.en.resources

6.2 AL.exe

Cette partie est un peu plus délicate.

Vous devez sélectionner tous les fichiers contenus dans un même répertoire pour obtenir à la fin une seule dll. Souvenez vous : un répertoire par langue, et dans un répertoire il y a un fichier ressource par WebForm.

Dans notre exemple, la génération de la dll pour l'Anglais "en" sera :

```
>al.exe /t:lib /embed:WebForm_A.en.resx  
/embed:WebForm_B.en.resx /culture:en  
/out:MyApp.resources.dll
```

Options:

/t: pour dire que vous voulez une dll
/embed: répéter cette option autant de fois qu'il y a de fichier à ajouter
/culture: cette option spécifie à quelle langue on associe cette dll
/out: le nom de la dll

Le nom de la dll doit toujours être de la forme : nom_d'application.resources.dll

6.3 Mettre les fichiers au bon endroit

Les dll doivent être placées de la façon suivante :

- Allez dans le répertoire BIN de votre application.
- Vous pouvez voir un répertoire par langue.
- S'il n'y a aucun répertoire, créez-en un pour chaque langue (toujours en suivant les conventions de nom)
- Mettez vos dll anglaises dans le répertoire "en".

- Mettez vos dll anglais-US dans le répertoire "en-us".
- Mettez vos dll français dans le répertoire "fr" etc...

Dans votre explorateur vous obtenez donc :



7. SOS ça marche pas !!! :

Il existe quelques outils pour vous aider à trouver et corriger le problème si vous en rencontrez un lors de la génération de la dll. Si en lançant votre application Web vous obtenez une erreur du genre "**Could not find any resources appropriate for the specified culture (or the neutral culture) in the given assembly.**" l'outil ildasm.exe peut vous aider. Cet outil va vous permettre de mieux visualiser le contenu d'une dll afin de trouver le problème.

Pour utiliser cet outil :
Comme pour les autres, ouvrez l'invite de commande Visual Studio .NET

Tapez:

```
> ildasm.exe votre_dll.dll
```

Je vous conseille de comparer votre dll générée par vos bons soins avec celle qui est générée par .NET à la compilation. Le problème le plus fréquent est quand vous comparez les deux champs ".mresource public". Si ces deux champs ne sont pas identiques, c'est que vous n'avez pas suivi les conventions de nom (et que vous avez donc lu de travers ce tutorial ;-))

8. La doc qui peut vous sauver la vie :

Toute cette documentation est en anglais malheureusement.

[Packaging and Deploying Resources](#)
[.NET Localization, Part 2: Creating Satellite Assemblies](#)
[Compiling Satellite Assemblies](#)
[Languages Naming Conventions](#)
[ResourceManager and ASP.NET](#)

Et je recommande tout simplement [le site MSDN de Microsoft](#).

Si malgré tous ces efforts vous vous arrachez encore les cheveux, posez votre question sur le forum .NET et nous nous ferons un plaisir de vous aider.

9. Conventions de noms, la liste :

Nom officiel - Code	Identifiant	Langue correspondante
af	0x0036	Afrikaans
af-ZA	0x0436	Afrikaans - South Africa
sq	0x001C	Albanian
sq-AL	0x041C	Albanian - Albania
ar	0x0001	Arabic
ar-DZ	0x1401	Arabic - Algeria
ar-BH	0x3C01	Arabic - Bahrain
ar-EG	0x0C01	Arabic - Egypt
ar-IQ	0x0801	Arabic - Iraq
ar-JO	0x2C01	Arabic - Jordan
ar-KW	0x3401	Arabic - Kuwait
ar-LB	0x3001	Arabic - Lebanon
ar-LY	0x1001	Arabic - Libya
ar-MA	0x1801	Arabic - Morocco
ar-OM	0x2001	Arabic - Oman
ar-QA	0x4001	Arabic - Qatar
ar-SA	0x0401	Arabic - Saudi Arabia
ar-SY	0x2801	Arabic - Syria
ar-TN	0x1C01	Arabic - Tunisia
ar-AE	0x3801	Arabic - United Arab Emirates
ar-YE	0x2401	Arabic - Yemen
hy	0x002B	Armenian
hy-AM	0x042B	Armenian - Armenia
az	0x002C	Azeri
Cy-az-AZ	0x082C	Azeri (Cyrillic) - Azerbaijan
Lt-az-AZ	0x042C	Azeri (Latin) - Azerbaijan
eu	0x002D	Basque
eu-ES	0x042D	Basque - Basque
be	0x0023	Belarusian
be-BY	0x0423	Belarusian - Belarus
bg	0x0002	Bulgarian
bg-BG	0x0402	Bulgarian - Bulgaria
ca	0x0003	Catalan
ca-ES	0x0403	Catalan - Catalan
zh-HK	0x0C04	Chinese - Hong Kong SAR
zh-MO	0x1404	Chinese - Macau SAR

zh-CN	0x0804	Chinese - China
zh-CHS	0x0004	Chinese (Simplified)
zh-SG	0x1004	Chinese - Singapore
zh-TW	0x0404	Chinese - Taiwan
zh-CHT	0x7C04	Chinese (Traditional)
hr	0x001A	Croatian
hr-HR	0x041A	Croatian - Croatia
cs	0x0005	Czech
cs-CZ	0x0405	Czech - Czech Republic
da	0x0006	Danish
da-DK	0x0406	Danish - Denmark
div	0x0065	Dhivehi
div-MV	0x0465	Dhivehi - Maldives
nl	0x0013	Dutch
nl-BE	0x0813	Dutch - Belgium
nl-NL	0x0413	Dutch - The Netherlands
en	0x0009	English
en-AU	0x0C09	English - Australia
en-BZ	0x2809	English - Belize
en-CA	0x1009	English - Canada
en-CB	0x2409	English - Caribbean
en-IE	0x1809	English - Ireland
en-JM	0x2009	English - Jamaica
en-NZ	0x1409	English - New Zealand
en-PH	0x3409	English - Philippines
en-ZA	0x1C09	English - South Africa
en-TT	0x2C09	English - Trinidad and Tobago
en-GB	0x0809	English - United Kingdom
en-US	0x0409	English - United States
en-ZW	0x3009	English - Zimbabwe
et	0x0025	Estonian
et-EE	0x0425	Estonian - Estonia
fo	0x0038	Faroese
fo-FO	0x0438	Faroese - Faroe Islands
fa	0x0029	Farsi
fa-IR	0x0429	Farsi - Iran
fi	0x000B	Finnish
fi-FI	0x040B	Finnish - Finland
fr	0x000C	French
fr-BE	0x080C	French - Belgium
fr-CA	0x0C0C	French - Canada
fr-FR	0x040C	French - France
fr-LU	0x140C	French - Luxembourg
fr-MC	0x180C	French - Monaco

fr-CH	0x100C	French - Switzerland
gl	0x0056	Galician
gl-ES	0x0456	Galician - Galician
ka	0x0037	Georgian
ka-GE	0x0437	Georgian - Georgia
de	0x0007	German
de-AT	0x0C07	German - Austria
de-DE	0x0407	German - Germany
de-LI	0x1407	German - Liechtenstein
de-LU	0x1007	German - Luxembourg
de-CH	0x0807	German - Switzerland
el	0x0008	Greek
el-GR	0x0408	Greek - Greece
gu	0x0047	Gujarati
gu-IN	0x0447	Gujarati - India
he	0x000D	Hebrew
he-IL	0x040D	Hebrew - Israel
hi	0x0039	Hindi
hi-IN	0x0439	Hindi - India
hu	0x000E	Hungarian
hu-HU	0x040E	Hungarian - Hungary
is	0x000F	Icelandic
is-IS	0x040F	Icelandic - Iceland
id	0x0021	Indonesian
id-ID	0x0421	Indonesian - Indonesia
it	0x0010	Italian
it-IT	0x0410	Italian - Italy
it-CH	0x0810	Italian - Switzerland
ja	0x0011	Japanese
ja-JP	0x0411	Japanese - Japan
kn	0x004B	Kannada
kn-IN	0x044B	Kannada - India
kk	0x003F	Kazakh
kk-KZ	0x043F	Kazakh - Kazakhstan
kok	0x0057	Konkani
kok-IN	0x0457	Konkani - India
ko	0x0012	Korean
ko-KR	0x0412	Korean - Korea
ky	0x0040	Kyrgyz
ky-KZ	0x0440	Kyrgyz - Kazakhstan
lv	0x0026	Latvian
lv-LV	0x0426	Latvian - Latvia
lt	0x0027	Lithuanian
lt-LT	0x0427	Lithuanian - Lithuania

mk	0x002F	Macedonian
mk-MK	0x042F	Macedonian - FYROM
ms	0x003E	Malay
ms-BN	0x083E	Malay - Brunei
ms-MY	0x043E	Malay - Malaysia
mr	0x004E	Marathi
mr-IN	0x044E	Marathi - India
mn	0x0050	Mongolian
mn-MN	0x0450	Mongolian - Mongolia
no	0x0014	Norwegian
nb-NO	0x0414	Norwegian (Bokmål) - Norway
nn-NO	0x0814	Norwegian (Nynorsk) - Norway
pl	0x0015	Polish
pl-PL	0x0415	Polish - Poland
pt	0x0016	Portuguese
pt-BR	0x0416	Portuguese - Brazil
pt-PT	0x0816	Portuguese - Portugal
pa	0x0046	Punjabi
pa-IN	0x0446	Punjabi - India
ro	0x0018	Romanian
ro-RO	0x0418	Romanian - Romania
ru	0x0019	Russian
ru-RU	0x0419	Russian - Russia
sa	0x004F	Sanskrit
sa-IN	0x044F	Sanskrit - India
Cy-sr-SP	0x0C1A	Serbian (Cyrillic) - Serbia
Lt-sr-SP	0x081A	Serbian (Latin) - Serbia
sk	0x001B	Slovak
sk-SK	0x041B	Slovak - Slovakia
sl	0x0024	Slovenian
sl-SI	0x0424	Slovenian - Slovenia
es	0x000A	Spanish
es-AR	0x2C0A	Spanish - Argentina
es-BO	0x400A	Spanish - Bolivia
es-CL	0x340A	Spanish - Chile
es-CO	0x240A	Spanish - Colombia
es-CR	0x140A	Spanish - Costa Rica
es-DO	0x1C0A	Spanish - Dominican Republic
es-EC	0x300A	Spanish - Ecuador
es-SV	0x440A	Spanish - El Salvador
es-GT	0x100A	Spanish - Guatemala
es-HN	0x480A	Spanish - Honduras

es-MX	0x080A	Spanish - Mexico
es-NI	0x4C0A	Spanish - Nicaragua
es-PA	0x180A	Spanish - Panama
es-PY	0x3C0A	Spanish - Paraguay
es-PE	0x280A	Spanish - Peru
es-PR	0x500A	Spanish - Puerto Rico
es-ES	0x0C0A	Spanish - Spain
es-UY	0x380A	Spanish - Uruguay
es-VE	0x200A	Spanish - Venezuela
sw	0x0041	Swahili
sw-KE	0x0441	Swahili - Kenya
sv	0x001D	Swedish
sv-FI	0x081D	Swedish - Finland
sv-SE	0x041D	Swedish - Sweden
syr	0x005A	Syriac
syr-SY	0x045A	Syriac - Syria
ta	0x0049	Tamil
ta-IN	0x0449	Tamil - India
tt	0x0044	Tatar
tt-RU	0x0444	Tatar - Russia
te	0x004A	Telugu
te-IN	0x044A	Telugu - India
th	0x001E	Thai
th-TH	0x041E	Thai - Thailand
tr	0x001F	Turkish
tr-TR	0x041F	Turkish - Turkey
uk	0x0022	Ukrainian
uk-UA	0x0422	Ukrainian - Ukraine
ur	0x0020	Urdu
ur-PK	0x0420	Urdu - Pakistan
uz	0x0043	Uzbek
Cy-uz-UZ	0x0843	Uzbek (Cyrillic) - Uzbekistan
Lt-uz-UZ	0x0443	Uzbek (Latin) - Uzbekistan
vi	0x002A	Vietnamese
vi-VN	0x042A	Vietnamese - Vietnam



L'ensemble ou partie de ce document ainsi que le code mis à disposition, ne peut être diffusé ailleurs sans autorisation préalable
elise.dupont@europe.com - www.dotnet-tech.com - 2003

