

Exercice : Le binary editor

1 Enoncé

L'objectif de cet exercice est double :

- De comprendre ce qu'est l'encoding en manipulant des chaînes de caractères et leur représentation binaire (via l'encoding justement)
 - D'implémenter un « Binary Editor ».
- Si vous ignorez ce qu'est un éditeur binaire, vous pouvez jeter un coup d'œil à celui qui est intégré dans Visual Studio de la façon suivante :
- Dans le menu : File -> Open -> File
 - Choisissez un fichier (de préférence un fichier de plusieurs centaines d'octets)
 - Cliquez sur la petite flèche à droite du bouton "Open" puis sur "Open With..."
 - Puis choisissez binary editor dans la liste

Vous devriez alors obtenir quelque chose ressemblant à ceci :

```

CryptIT.sln
00000000 EF BB 8F 0D 0A 4D 69 63 72 6F 73 6F 66 74 20 56 ....Microsoft V
00000010 69 73 75 61 6C 20 53 74 75 64 69 6F 20 53 6F 6C isual Studio Sol
00000020 75 74 69 6F 6E 20 46 69 6C 65 2C 20 46 6F 72 6D ution File, Form
00000030 61 74 20 56 65 72 73 69 6F 6E 20 31 32 2E 30 30 at Version 12.00
00000040 0D 0A 23 20 56 69 73 75 61 6C 20 53 74 75 64 69 ..# Visual Studi
00000050 6F 20 32 30 31 33 0D 0A 56 69 73 75 61 6C 53 74 o 2013..VisualSt
00000060 75 64 69 6F 56 65 72 73 69 6F 6E 20 3D 20 31 32 udioVersion = 12
00000070 2E 30 2E 33 30 35 30 31 2E 30 0D 0A 4D 69 6E 69 .0.30501.0..Mini
00000080 6D 75 6D 56 69 73 75 61 6C 53 74 75 64 69 6F 56 mumVisualStudioV
00000090 65 72 73 69 6F 6E 20 3D 20 31 30 2E 30 2E 34 30 ersion = 10.0.40
000000a0 32 31 39 2E 31 0D 0A 50 72 6F 6A 65 63 74 28 22 219.1..Project("
000000b0 78 46 41 45 30 34 45 43 30 2D 33 30 31 46 2D 31 {FAE04EC0-301F-1
000000c0 31 44 33 2D 42 46 34 42 2D 30 30 43 30 34 46 37 1D3-BF4B-00C04F7
000000d0 39 45 46 42 43 7D 22 29 20 3D 20 22 43 72 79 70 9EFBC}") = "Cryp
000000e0 74 49 54 22 2C 20 22 43 72 79 70 74 49 54 5C 43 tIT", "CryptIT\C
000000f0 72 79 70 74 49 54 2E 63 73 70 72 6F 6A 22 2C 20 rypIT.csproj",
00000100 22 78 46 41 33 46 46 46 43 46 2D 31 42 30 32 2D "{FA3FFFCF-1B02-
00000110 34 45 45 41 2D 38 34 46 34 2D 42 37 30 30 35 42 4EEA-84F4-B7005B
00000120 35 39 31 37 42 30 7D 22 0D 0A 45 6E 64 50 72 6F 5917B0}"..EndPro
00000130 6A 65 63 74 0D 0A 50 72 6F 6A 65 63 74 28 22 7B ject..Project("{
00000140 46 41 45 30 34 45 43 30 2D 33 30 31 46 2D 31 31 FAE04EC0-301F-11
00000150 44 33 2D 42 46 34 42 2D 30 30 43 30 34 46 37 39 D3-BF4B-00C04F79
00000160 45 46 42 43 7D 22 29 20 3D 20 22 49 54 2E 43 72 EFBC}") = "IT.Cr
00000170 79 70 74 49 54 22 2C 20 22 49 54 2E 43 72 79 70 yptIT", "IT.Cryp
00000180 74 49 54 5C 49 54 2E 43 72 79 70 74 49 54 2E 63 tIT\IT.CryptIT.c
  
```

L'éditeur est organisé en trois zones, avec de gauche à droite :

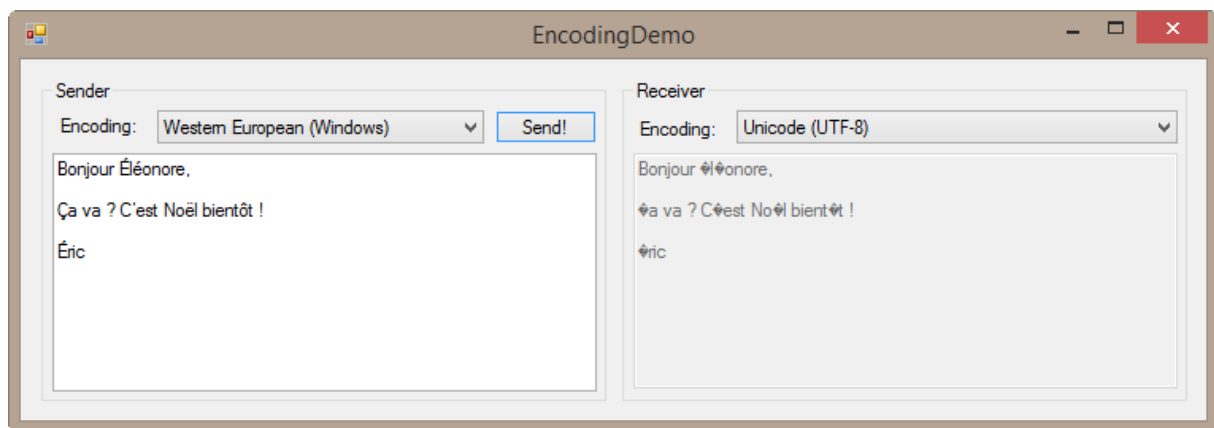
- Les **adresses** (ou offsets) en hexadécimal
C'est l'adresse du premier octet de chaque ligne. Par exemple, à la troisième ligne, on peut lire la valeur 00000020. Ce qui signifie que le premier octet de la troisième ligne (qui a une valeur de 75) est situé à l'adresse 20 dans le fichier (soit 32 en base 10, cet octet est donc le 33^{ème} dans le fichier, le premier octet étant situé à l'adresse 0).
- Le **contenu** du fichier en hexadécimal
Chaque octet du fichier est simplement lu et représenté par deux chiffres hexadécimaux (on rappelle qu'un octet peut contenir un nombre de 0 à 255, et que deux chiffres hexadécimaux permettent d'écrire des nombre de 00 à FF ce qui correspond bien à 0 et 255 en base 10). Le premier octet du fichier ouvert dans l'exemple vaut EF, le 10^{ème} vaut 6F.
- Une représentation **ASCII** du contenu
La zone de droite contient une représentation ASCII du fichier ouvert. Pour chaque octet, l'éditeur tente d'afficher le caractère ASCII correspondant. L'éditeur n'affiche que les caractères ASCII lisibles, c'est à dire, les lettres non accentuées majuscules et minuscules, les

chiffres et quelques symboles comme les accolades, les parenthèses ou la ponctuation. Les octets qui ne sont pas associés à un caractère ASCII lisible sont représentés par un point. Dans cet exemple on peut voir que les 5 premiers caractères ne correspondent à aucun caractère ASCII lisible. Le 6ème en revanche (avec la valeur 4D) correspond au caractère 'M'.

Implémenter un binary editor n'est pas trivial. Nous allons commencer par quelque chose de plus simple et ajouter des fonctionnalités au fur et à mesure.

2 Itération 1 : Un encoder / décodeur simple

Dans cette itération, l'objectif est d'implémenter une application ressemblant à ceci :



Cette application s'utilise de la façon suivante :

1. L'utilisateur saisi un texte dans la TextBox situé dans la zone "sender"
2. Il choisit un encoding (parmi tous les encodings disponibles sur la machine) dans la ComboBox située dans la zone "sender"
3. Il choisit un encoding dans la ComboBox située dans la zone "Receiver"
4. Il clique sur le bouton "Send!"

L'application doit alors :

1. Construire un tableau d'octets à partir du texte saisi et de l'encoding spécifié dans la zone "Sender"
2. Décoder ce tableau d'octets en utilisant l'encoding spécifié dans la zone "Receiver"
3. Afficher la chaîne ainsi obtenue dans la zone de texte située dans la zone "Receiver"

Note : Bien entendu, il est normal dans certains cas que le texte du receiver soit illisible. Et c'est bien le but de cet exercice : comprendre ce qu'est l'encoding et pourquoi, des fois, on reçoit des mails avec des caractères bizarres dedans... c'est lorsque l'encoding de départ n'est pas le même que celui d'arrivée !

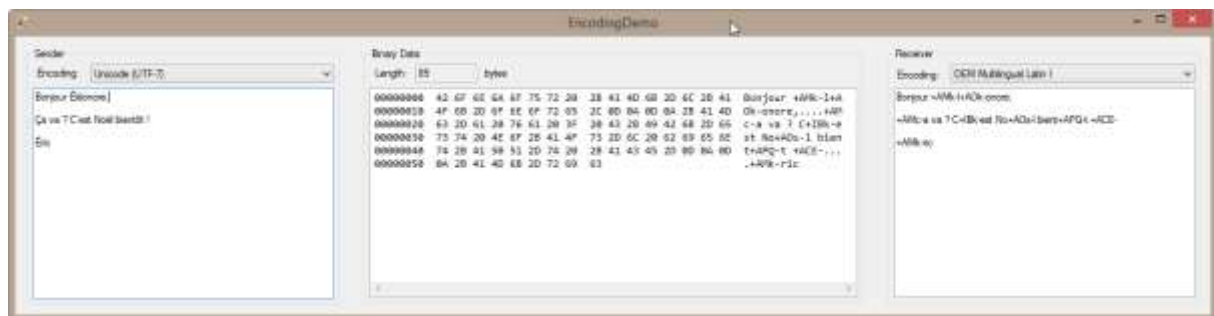
3 Itération 2 : Suppression du bouton "Send!"

Supprimez le bouton "Send!" et faites en sorte que le receiver soit mis à jour, dès que la zone de texte est modifiée dans le sender.

4 Itération 3 : Ajout d'un décodeur binaire

Nous allons maintenant ajouter un deuxième décodeur, qui affichera le tableau d'octet envoyé directement en hexadécimal, à la manière de l'éditeur binaire de Visual Studio.

L'idée est d'obtenir quelque chose ressemblant à ceci :



5 Itération 4 : Transformer le binary reader en binary editor

Modifiez maintenant votre application pour rendre la zone de texte dans la zone "binary data" éditable. A chaque modification d'octet (dans la zone centrale), la zone de texte dans la zone "Receiver" doit être mise à jour.