

```

1  using Microsoft.VisualBasic.FileIO;
2  using System;
3  using System.Collections.Generic;
4  using System.ComponentModel;
5  using System.Data;
6  using System.Drawing;
7  using System.IO;
8  using System.Linq;
9  using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;
12 using System.Xml.Linq;
13
14
15 namespace TP1
16 {
17     public partial class Form1 : Form
18     {
19         public Form1()
20         {
21             InitializeComponent();
22             LoadTreeView();
23         }
24
25         private void ListeRepertoire(TreeNode _node)
26         {
27             _node.Nodes.Clear(); // vide la collection de Nodes de _node
28             DirectoryInfo di = new DirectoryInfo(_node.Tag.ToString()); // Pointe
29             sur le Drive ou Répertoire du node
30             foreach (DirectoryInfo dif in di.GetDirectories()) // parcours tous les
31             répertoires du parent
32             {
33                 try {
34                     TreeNode tn = new TreeNode(dif.Name); // Créer un node avec le
35                     nom du répertoire
36                     tn.ImageIndex = 4; // Affect l'i dice de
37                     l'image
38                     tn.Tag = dif.FullName; // charge le chemin du
39                     répertoire
40                     _node.Nodes.Add(tn); // ajoute ce nouveau
41                     note comme enfant du node passé en paramètre
42
43                     if (dif.GetDirectories().Length > 0) // indique si il y a des
44                     sous répertoires dans "dif"
45                     {
46                         tn.Nodes.Add(""); // ajoute un node vide pour
47                         avoir un "+" devant le node
48                     }
49                 }
50                 catch { }
51             }
52         }
53         private void treeView1_BeforeExpand(object sender, TreeViewCancelEventArgs e)
54         {
55             if (e.Node.Level > 1) // évite le node racine et les lecteurs (déjà
56             chargés)
57             {
58                 ListeRepertoire(e.Node); // e.Node = le node sur lequel
59                 l'utilisateur a cliqué sur +
60             }
61         }
62         private void treeView1_AfterSelect(object sender, TreeViewEventArgs e)
63         {
64             // Événement déclenché après la sélection d'un node
65             if (e.Node.Level > 0) // évite le node racine qui n'a pas de fichiers
66             {
67                 LoadListView(e.Node.Tag.ToString());
68             }
69         }
70         private void LoadListView(String sPathDirectory)
71         {
72             listView1.Items.Clear(); // Vide le contenu de la listview
73             DirectoryInfo di = new DirectoryInfo(sPathDirectory); // e.Node = node

```

```

64         sélectionné, Tag son chemin
65         try
66         { // cas où votre application n'a pas les droits d'accès au
        répertoire ou à certains fichiers
67             foreach (FileInfo fi in di.GetFiles()) // Parcourt tous les
        fichiers du répertoire fi
68             {
69                 int i = 0;
70                 // listView1 = la listview qui va afficher les fichiers,
        elle a une collection d'items
71                 if (fi.Name.Substring(fi.Name.LastIndexOf(".")).ToLower() ==
        ".pdf")
72                 {
73                     i = 1; // indice de l'icon pdf
74                 }
75                 listView1.Items.Add(fi.Name, 1); // 0 => index de l'icon
        dans l'imagelist
76
77                 // Autre façon d'affecter le bon icon
78                 String ext = System.IO.Path.GetExtension(fi.Name); //
        récupère l'extension du fichier
79                 if (ext.ToLower() == ".pdf") { i = 1; }
80                 if ((ext.ToLower() == ".doc") || (ext.ToLower() == ".docx"))
        { i = 2; }
81                 if ((ext.ToLower() == ".xls") || (ext.ToLower() == ".xlsx"))
        { i = 3; }
82                 if ((ext.ToLower() == ".jpg") || (ext.ToLower() == ".jpeg"))
        { i = 4; }
83                 if (ext.ToLower() == ".txt") { i = 5; }
84                 listView1.Items[listView1.Items.Count - 1].ImageIndex = i;
        // listView1.Items.Count - 1 => nombre total d'items - 1 =
        indice du dernier item ajouté
85             }
86         }
87         catch { }
88     }
89     private void listView1_SelectedIndexChanged(object sender, EventArgs e)
90     {
91         try
92         {
93             if (listView1.SelectedItems.Count > 0) // il y a au moins un item de
        sélectionné
94             {
95                 String sPath = treeView1.SelectedNode.Tag.ToString(); //
        répertoire du fichier
96                 sPath = sPath + @"\" + listView1.SelectedItems[0].Text; //
        nom du fichier (1° item sélectionné => indice 0
97
98                 label1.Text = "Emplacement : " + sPath;
99                 // textBox1.text = sPath;
100                 FileInfo fi = new FileInfo(sPath);
101                 label2.Text = "Date de dernière modification : " +
        fi.LastWriteTime.ToString("dd/MM/yyyy hh:mm");
102                 label3.Text = "Taille : ";
103                 double dSize = fi.Length;
104                 if (dSize < 1024) { label3.Text += dSize.ToString("0") + "
        octets"; }
105                 else
106                 {
107                     dSize = dSize / 1024;
108                     if (dSize < 1024) { label3.Text += dSize.ToString("0.00") +
        " ko"; }
109                     else
110                     {
111                         dSize = dSize / 1024;
112                         if (dSize < 1024) { label3.Text +=
        dSize.ToString("0.00") + " Mo"; }
113                         else
114                         {
115                             dSize = dSize / 1024;
116                             label3.Text += dSize.ToString("0.00") + " Go";
117                         }
118                     }
119                 }
120             }
121         }
122     }

```

```

119         }
120     }
121 }
122     catch { }
123 }
124 private void LoadTreeView()
125 {
126     treeView1.Nodes.Add("Racine", "Racine", 0); // Key, text, indice icon
127     image list
128     foreach (DriveInfo di in DriveInfo.GetDrives()) // Parcours les lecteurs
129         (drive)
130     {
131         try
132         {
133             //treeView1.Nodes[0].Nodes.Add(di.Name, );
134             // Autre façon
135             TreeNode tn = new TreeNode(di.Name); // Ajoute un node avec
136             le nom du lecteur
137             tn.Tag = di.Name; // Ajoute le chemin à Tag
138             if (di.DriveType == DriveType.Fixed) // test si DD
139             {
140                 tn.ImageIndex = 1; // index de l'image dans l'imagelist
141                 affecté au TrreView
142             }
143             if (di.DriveType == DriveType.Removable) // si lecteur Usb
144             {
145                 tn.ImageIndex = 2; // n'utilise pas le même index, donc pas
146                 la même image
147             }
148             if (di.DriveType == DriveType.Network) // si lecteur Network
149             {
150                 tn.ImageIndex = 3;
151             }
152             treeView1.Nodes[0].Nodes.Add(tn);
153             ListeRepertoire(tn);
154         }
155     }
156     catch { }
157 }
158
159 private void radioButton1_CheckedChanged(object sender, EventArgs e)
160 {
161     if (radioButton1.Checked) { listView1.View = View.List; }
162 }
163
164 private void radioButton2_CheckedChanged(object sender, EventArgs e)
165 {
166     if (radioButton2.Checked) { listView1.View = View.SmallIcon; }
167 }
168
169 private void radioButton3_CheckedChanged(object sender, EventArgs e)
170 {
171     if (radioButton3.Checked) { listView1.View = View.LargeIcon; }
172 }
173
174 private void quitterToolStripMenuItem_Click(object sender, EventArgs e)
175 {
176     this.Close();
177 }
178
179 private void copierToolStripMenuItem_Click(object sender, EventArgs e)
180 {
181     try
182     {
183         System.Collections.Specialized.StringCollection sc =
184             new
185                 System.Collections.Specialized.StringColle
186                 ction(); // Collection de chaine
187         String sPath = treeView1.SelectedNode.Tag.ToString() + @"\" +
188             listView1.SelectedItems[0].Text;
189         sc.Add(sPath); // ajoute le fichier à la collection
190         Clipboard.SetFileDropList(sc); // passe la collection au Clipboard
191     }
192 }

```

```

184         catch { }
185     }
186     private void collerToolStripMenuItem_Click(object sender, EventArgs e)
187     {
188         try
189         {
190             System.Collections.Specialized.StringCollection sc
191                 = Clipboard.GetFilesDropList();
192                 // récupère la collection du
193                 // Clipboard
194             String sPathOrigine = sc[0]; // fichier d'origine exemple :
195             c:\temp\Monfichier.doc
196             String sNomFichier =
197             sPathOrigine.Substring(sPathOrigine.LastIndexOf(@"\")); // Extrait
198             Monfichier.doc de sPathOrigine
199             String sPathDestination = treeView1.SelectedNode.Tag.ToString() +
200             @"\" + sNomFichier; // c:\Destination\Monfichier.doc
201             if (File.Exists(sPathDestination)) // cas où le fichiers destination
202             exist
203             {
204                 sPathDestination = sPathDestination.Substring(0,
205                 sPathOrigine.LastIndexOf(@".") + 1) +
206                 " - Copie" +
207
208                 sPathDestination.Substring(sPathOrigine.LastIn
209                 dexOf(@".") + 1);
210
211                 // change c:\Destination\Monfichier.doc en
212                 c:\Destination\Monfichier - Copie.doc
213             }
214             File.Copy(sPathOrigine, sPathDestination, true); // copy le fichier
215             LoadListView(treeView1.SelectedNode.Tag.ToString()); // recharge la
216             listView
217             //MessageBox.Show(sc[0]);
218         }
219         catch { }
220     }
221     private void supprimerToolStripMenuItem_Click(object sender, EventArgs e)
222     {
223         try
224         {
225             String sPath = treeView1.SelectedNode.Tag.ToString() + @"\" +
226             listView1.SelectedItems[0].Text;
227             if (MessageBox.Show("Voulez vous supprimer ce fichier ?" +
228             Environment.NewLine + sPath,
229             "Attention", MessageBoxButtons.OKCancel) ==
230             DialogResult.OK)
231             {
232                 //File.Delete(sPath); // Efacement Physique
233                 FileSystem.DeleteFile(sPath, UIOption.AllDialogs,
234                 RecycleOption.SendToRecycleBin,
235
236                 UICancelOption.ThrowExcept
237                 ion); // Force la
238                 corbeille
239                 // Nécessite using Microsoft.VisualBasic.FileIO;
240
241                 LoadListView(treeView1.SelectedNode.Tag.ToString()); //
242                 Rafraichit la listView
243
244                 // File.Move(sPath, "C:\\$Recycle.Bin\\" +
245                 listView1.SelectedItems[0].Text); // Ne marche pas sans
246                 autorisation
247             }
248         }
249         catch (Exception ex) // Si erreur charge la structure Exception avec
250         les infos de l'erreur
251         {
252             MessageBox.Show(ex.Message); // Affiche le message d'erreur renvoyé
253         }
254     }
255     private void listView1_MouseDown(object sender, MouseEventArgs e) // Appelé
256     lorsqu'un bouton de la souris est enfoncé
257     {

```

```

233     if (e.Button == MouseButton.Right) // tst si il s'agit du bouton droit
234     {
235         // Tester si la souris pointe sur un item
236         ListViewHitTestInfo hitTestInfo = listView1.HitTest(e.X, e.Y);
237         if (hitTestInfo.Item == null) // Pas d'item => le curseur de la
            souris n'est pas sur un item
238         {
239             //MessageBox.Show("Pas d'item");
240             contextMenuStrip2.Show(listView1, e.X, e.Y);
241         }
242         else // non null => curseur de la souris est sur un item
243         {
244             //MessageBox.Show("item");
245             contextMenuStrip1.Show(listView1, e.X, e.Y); // affiche le menu
            sur la listView1 aux coordonnées e.X et e.Y
246         }
247     }
248 }
249
250 private void listView1_DragEnter(object sender, DragEventArgs e)
251 {
252     //
253 }
254
255 private void listView1_ItemDrag(object sender, ItemDragEventArgs e)
256 { // Événement appelé au démarrage de l'opération de DragAndDrop
257     listView1.DoDragDrop(e.Item, DragDropEffects.Move);
258     // DoDragDrop => prévient le système de démarrer l'opération de
259     DragAndDrop
260     // e.Item => l'item sélectionné
261     // DragDropEffects.Move => opération de déplacement
262 }
263
264 private void listView1_DragOver(object sender, DragEventArgs e)
265 { // Événement appelé si la souris survole un composant pendant une
266     opération de DragAndDrop
267     e.Effect = DragDropEffects.Move; // change le curseur de la souris
268 }
269
270 private void treeView1_DragOver(object sender, DragEventArgs e)
271 {
272     e.Effect = DragDropEffects.Move;
273 }
274
275 private void treeView1_DragDrop(object sender, DragEventArgs e)
276 {
277     // Trouver l'item de départ
278     // e.Data contient les données transférées, et typeof(ListViewItem)
279     // permet de spécifier que c'est un ListViewItem
280     ListViewItem lvi_move =
281         (ListViewItem)e.Data.GetData(typeof(ListViewItem));
282
283     // Trouver le node d'arrivée
284     // les coordonnées sont connues dans "e" et
285     // GetNodeAt permet de donner le node en fonction du point d'arrivée
286     TreeNode node_destination =
287         treeView1.GetNodeAt(treeView1.PointToClient(new Point(e.X, e.Y)));
288
289     if (node_destination != null)
290     {
291         // Trouver le chemin du fichier de départ et le chemin de fichier
292         d'arrivée
293         String sNomFichier = lvi_move.Text;
294         String sPathSource = treeView1.SelectedNode.Tag.ToString() + @"\" +
295             sNomFichier;
296         String sPathDestination = node_destination.Tag.ToString() + @"\" +
297             sNomFichier;
298         // Déplacer le fichier physiquement
299         try
300         {
301             File.Move(sPathSource, sPathDestination);
302             LoadListView(treeView1.SelectedNode.Tag.ToString());
303             MessageBox.Show("Fichier déplacé dans " + sPathDestination,

```

```
297         "Info", MessageBoxButtons.OK, MessageBoxIcon.Information);
298     }
299     catch { }
300 }
301
302
303
304     }
305 }
306 }
307
```