

# Les fichiers

Création, écriture et lecture des fichiers sous DELPHI

**Ricco Rakotomalala**  
**Université Lumière Lyon 2**

# Fichiers ?

**Fichier** - Collection d'informations sur une mémoire de masse (non volatile, capacité de stockage plus importante que la mémoire vive).

**Types de fichiers** - Les fichiers se distinguent par :

1. **Organisation des données** (structuré = fichiers d'enregistrements, plus contraignants, éditable uniquement par les outils respectant à la lettre le format ; non structuré = fichiers textes, plus souples, éditables en dehors du logiciel).
2. **Accès aux données** (accès indexé, utilisable comme un tableau ; séquentiel, lecture des informations ligne par ligne).



Il est possible de manipuler les données sans avoir à les charger dans leur totalité en mémoire.

Non structuré, accès séquentiel

# **FICHER TEXTE**

# Fichier texte DELPHI : utilisation du type prédéfini **TEXTFILE**

Non structuré

Accès séquentiel, ligne par ligne, pas de possibilité de retour en arrière

Chaque ligne = une chaîne de caractère, quelle que soit son expression

Pas de possibilité de remplacement d'une ligne

Mais possibilité d'ouverture en ajout

## 2 étapes obligatoires pour la manipulation d'un fichier

//déclaration d'un variable permettant de manipuler un fichier texte

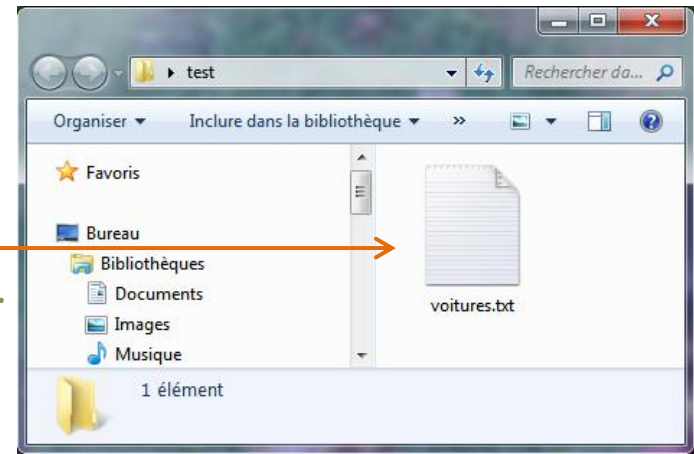
Var **F**: **TextFile**;

//associer la variable au fichier physique présent sur le disque

//'voitures.txt' représente le nom du fichier

**AssignFile(F, 'voitures.txt');**

Non de fichier peut être une variable saisie par l'utilisateur.  
Pour l'instant, il le cherche dans le dossier du projet.  
Si le fichier n'existe pas, pas de plantage à ce stade.



## Fichier texte DELPHI : ouverture en création et en écriture

```
Project1.dpr
Project1
program Project1;
{$APPTYPE CONSOLE}
uses
  SysUtils;
var F: TextFile;
    ligne: string;
begin
  //association F <-> Fichier physique
  AssignFile(F, 'voitures.txt');
  //ouverture en écriture
  Rewrite(F);
  //écriture ligne à ligne
  //passage par une variable
  ligne:= 'clio';
  Writeln(F, ligne);
  //écriture ligne directement
  Writeln(F, 'passat');
  Writeln(F, 'c4');
  Writeln(F, 'corsa');
  Writeln(F, 'megane');
  //fermeture
  CloseFile(F);
  //
  readln;
end.
```

//ouverture :  
//si le fichier n'existe pas => création  
//si le fichier existe => écrasement

**Rewrite(F);**

//écriture d'une ligne

**Writeln(F, chaîne de carac.);**

//fermeture du fichier

**CloseFile(F);**



```
voitures.txt - Bloc-notes
Fichier  Edition  Format  Affichage  ?
clio
passat
c4
corsa
megane
|
```

Le fichier est consultable / modifiable, dans tout éditeur de texte. Très grande souplesse d'utilisation.

# Fichier texte DELPHI : ouverture en lecture

```
Project2.dpr
Project2
program Project2;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var F: TextFile;
    ligne: string;
    somme, valeur: double;

begin
  //branchement
  AssignFile(F, 'nombres.txt');
  //ouverture en lecture
  Reset(F);
  //lecture de chaque ligne
  //tant que fin de fichier non atteinte
  //addition des valeurs
  somme:= 0.0;
  while (Eof(F) = false) do
  begin
    //lecture d'une ligne
    Readln(F, ligne);
    //conversion en numérique
    valeur:= StrToFloat(ligne);
    //addition
    somme:= somme + valeur;
  end;
  //fermeture du fichier
  CloseFile(F);
  //affichage de la somme
  writeln('Somme = ', somme);
  readln;
end.
```

//ouverture :

//si le fichier n'existe pas => erreur

//si le fichier existe => Ok

**Reset(F);**

//lecture d'une ligne

**Readln(F, chaîne de carac.);**

//pour savoir si on est à la fin du fichier

**Eof(F);** //renvoie true ou false

//fermeture du fichier

**CloseFile(F);**



```
nombres.txt - Bloc-notes
Fichier  Edition  Format  Affichage ?
15
17
19
16
8
6
43
```



```
C:\Users\Maison\Desktop\test\P...
Somme = 1.2400000000000000E+0002
```

## Fichier texte DELPHI : ouverture en ajout (après la dernière ligne)

```
Project3.dpr
Project3
program Project3;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var F: TextFile;
begin
  //association F <-> Fichier physique
  AssignFile(F, 'voitures.txt');
  //ouverture en écriture
  Append(F);
  //écriture ligne directement
  Writeln(F, 'cayenne');
  //fermeture
  CloseFile(F);
  readln;
end.
```

//ouverture :  
//si le fichier n'existe pas => erreur  
//si le fichier existe => OK

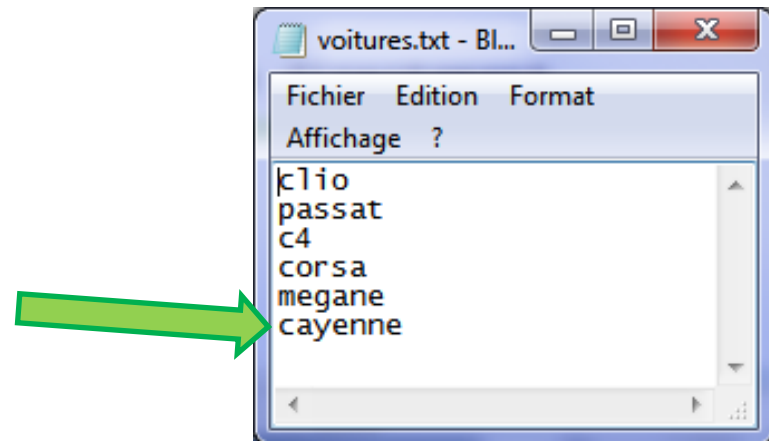
**Append(F);**

//écriture d'une ligne

**Writeln(F, chaîne de caract.);**

//fermeture du fichier

**CloseFile(F);**



Fichier d'enregistrements

Accès direct (indiqué)

**FICHER TYPÉ**



## Fichiers typés (binaires) DELPHI : fichiers d'enregistrements **record, file of record**

Structuré : chaque valeur correspond à un enregistrement

Accès direct : possibilité d'accéder à un enregistrement via son indice

- Possibilité de remplacer un enregistrement existant
- Possibilité d'ajouter un enregistrement
- Possibilité de tronquer le fichier (supprimer des enregistrements situés en fin de fichier)

//déclaration d'un type enregistrement

TYPE TMaison = **record**

    quartier : shortstring;

    surface : double;

**end;**

4 étapes obligatoires pour  
la manipulation d'un fichier  
d'enregistrements

//déclaration d'un type fichier d'enregistrements

TYPE TFileMaison = **file of** TMaison;

//déclaration d'un variable permettant de manipuler ce type de fichier

Var **F**: TFileMaison;

//associer la variable au fichier physique présent sur le disque

//'maisons.dat' représente le nom du fichier (on fixe l'extension à notre guise)

AssignFile(**F**, 'maisons.dat');

# Fichier typé DELPHI : ouverture en création et en écriture

```
Project4.dpr
Project4
program Project4;
{$APPTYPE CONSOLE}
uses
  SysUtils;
TYPE
  TMaison = record
    quartier: shortstring;
    surface: double;
  end;
  TFileMaison = file of TMaison;
var
  F: TFileMaison;
  maison: TMaison;
begin
  //maison
  maison.quartier:= 'bron';
  maison.surface:= 150;
  //association F avec fichier physique
  AssignFile(F, 'maisons.dat');
  //ouverture en écriture
  Rewrite(F);
  //écriture en 1ere position
  write(F,maison);
  //pointeur de fichier en fin de fichier
  maison.quartier:= 'saint priest';
  maison.surface:= 234;
  write(F,maison);
  //fermeture fichier
  CloseFile(F);
  readln;
end.
```

//ouverture :  
//si le fichier n'existe pas => création  
//si le fichier existe => écrasement

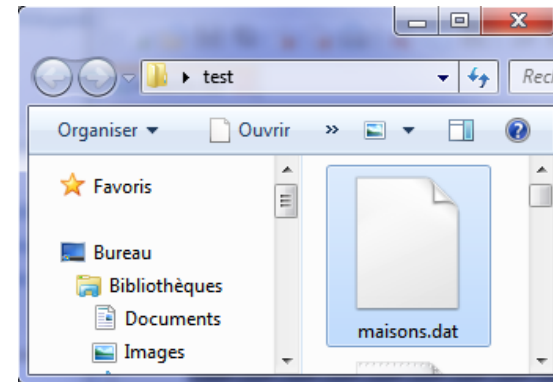
**Rewrite(F);**

//écriture d'un enregistrement

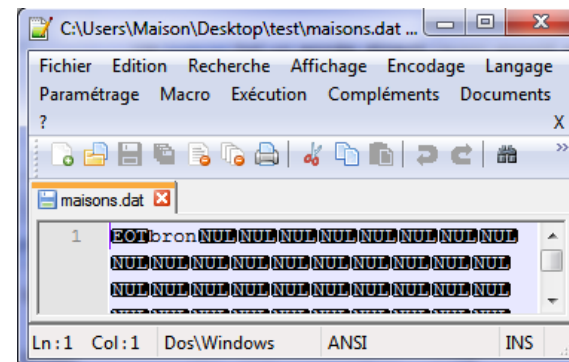
**Write (F, variable);**

//fermeture du fichier

**CloseFile(F);**



Le fichier est bien créé, mais il n'est pas consultable avec un éditeur de texte.



# Fichier typé DELPHI : ouverture en lecture

```
Project5.dpr
Project5
program Project5;

{$APPTYPE CONSOLE}

uses
  SysUtils;

TYPE
  TMaison = record
    quartier: shortstring;
    surface: double;
  end;

  TFileMaison = file of TMaison;

var F: TFileMaison;
    maison: TMaison;
begin
  //association F avec fichier physique
  AssignFile(F, 'maisons.dat');
  //ouverture en lecture
  Reset(F);
  //tant que pas fin de fichier
  while (eof(f) = false) do
  begin
    //lire un enregistrement
    Read(F, maison);
    //affichage à l'écran
    writeln('quartier = ', maison.quartier);
    writeln('surface = ', maison.surface);
  end;
  //fermeture fichier
  CloseFile(F);
  readln;
end.
```

//ouverture :  
//si le fichier n'existe pas => erreur  
//si le fichier existe => Ok  
//positionnement sur le 1<sup>er</sup> enregistrement  
**Reset(F);**  
//lecture d'un enregistrement  
//la lecture provoque le passage au suivant  
**Read (F, enregistrement);**  
//pour savoir si on est à la fin du fichier  
**Eof(F);** //renvoie true ou false  
//fermeture du fichier  
**CloseFile(F);**



```
C:\Users\Maison\Desktop\test\Project5.exe
quartier = bron
surface = 1.5000000000000000E+0002
quartier = saint priest
surface = 2.3400000000000000E+0002
```

```
program Project6;
```

```
{$APPTYPE CONSOLE}
```

```
uses
```

```
  SysUtils;
```

```
TYPE
```

```
  TMaison = record  
    quartier: shortstring;  
    surface: double;  
  end;
```

```
  TFileMaison = file of TMaison;
```

```
var F: TFileMaison;
```

```
    maison: TMaison;
```

```
begin
```

```
  //association F avec fichier physique
```

```
  AssignFile(F, 'maisons.dat');
```

```
  //ouverture en lecture/modification
```

```
  Reset(F);
```

```
  //se positionner sur le 2nd enregistrement
```

```
  //modifier sa surface
```

```
  Seek(F, 1);
```

```
  Read(F, maison); //attention, modifie la position
```

```
  maison.surface:= 500;
```

```
  //se repositionner sur le bon enregistrement
```

```
  Seek(F, 1);
```

```
  Write(F, maison);
```

```
  //afficher toutes les maisons
```

```
  Seek(F, 0);
```

```
  while (eof(f) = false) do
```

```
  begin
```

```
    //lire un enregistrement
```

```
    Read(F, maison);
```

```
    //affichage à l'écran
```

```
    writeln('quartier = ', maison.quartier);
```

```
    writeln('surface = ', maison.surface);
```

```
  end;
```

```
  //fermeture fichier
```

```
  CloseFile(F);
```

```
  readln;
```

```
end.
```

## Fichier typé DELPHI : ouverture en lecture / modification

//se positionner sur l'enregistrement n°i

//le 1<sup>er</sup> enregistrement a le numéro 0

**Seek(F, i);**

//Position actuelle du curseur de fichier

**FilePos (F);**

//Nombre d'enregistrements dans le fichier

**FileSize(F);**

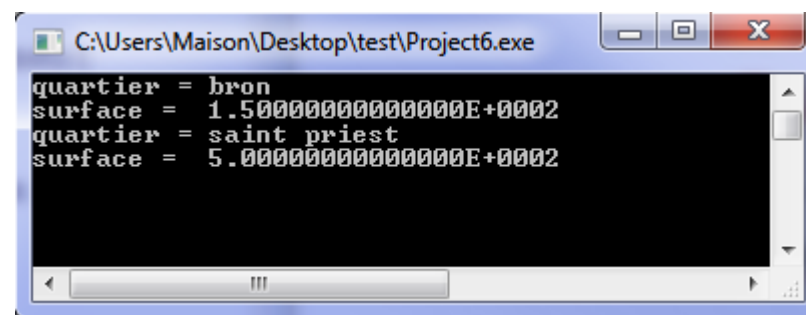
//supprime les enregistrements situés à partir

//de la position actuelle du curseur

**Truncate(F);**

Ex. : Seek(F, 0); Truncate(F); //vide le fichier

Ex. : Seek(F, FileSize(F)); //emmène à la fin du fichier

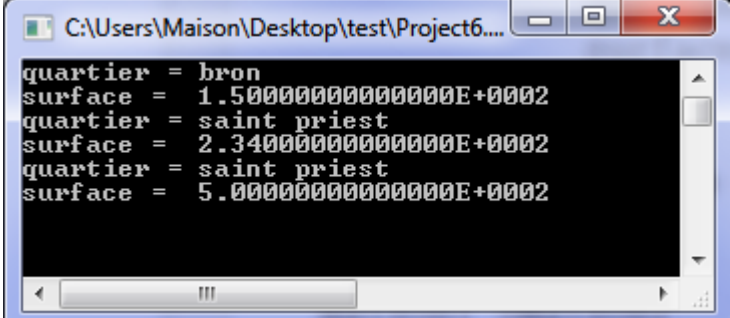


```
C:\Users\Maison\Desktop\test\Project6.exe  
quartier = bron  
surface = 1.5000000000000000E+0002  
quartier = saint priest  
surface = 5.0000000000000000E+0002
```

Un fichier typé peut s'utiliser comme un tableau. **Mais il est bien plus lent.**

Fichier typé DELPHI : ouverture en lecture / modification – **Que se passe-t-il si on oublie le repositionnement ?**

```
var F: TFileMaison;
    maison: TMaison;
begin
    //association F avec fichier physique
    AssignFile(F, 'maisons.dat');
    //ouverture en lecture/modification
    Reset(F);
    //se positionner sur le 2nd enregistrement
    //modifier sa surface
    Seek(F, 1);
    Read(F, maison); //attention, modifie la position du curseur
    maison.surface:= 500;
    //se repositionner sur le bon enregistrement
    //Seek(F, 1); ←
    Write(F, maison);
    //afficher toutes les maisons
    Seek(F, 0);
    while (eof(f) = false) do
    begin
        //lire un enregistrement
        Read(F, maison);
        //affichage à l'écran
        writeln('quartier = ', maison.quartier);
        writeln('surface = ', maison.surface);
    end;
    //fermeture fichier
    CloseFile(F);
    readln;
end.
```



```
C:\Users\Maison\Desktop\test\Project6...
quartier = bron
surface = 1.5000000000000000E+0002
quartier = saint priest
surface = 2.3400000000000000E+0002
quartier = saint priest
surface = 5.0000000000000000E+0002
```

L'enregistrement est inscrit à la fin du fichier (ou sur l'enregistrement suivant s'il existe !!!). **Danger !**

# FIN...

Les mêmes concepts sont – à peu de choses près – présents dans tous les langages de programmation...