

Windows Phone

Persistence de données locale

IsolatedStorage

- IsolatedStorage : un espace privé réservé à une application dans le système de fichier et inaccessible aux autres applications.
- Espace de noms: System.IO.IsolatedStorage
- Deux classes principales
 - IsolatedStorageSettings : implémente une structure de dictionnaire (clé, valeurs)
 - IsolatedStorageFile :
 - Une classe qui représente la zone de stockage.
 - Supporte des méthodes pour créer et ouvrir des fichiers et des dossiers

IsolatedStorageFile

utilisation

```
IsolatedStorageFile store =  
IsolatedStorageFile.GetUserStoreForApplication();  
    //Traitement  
store.Dispose();
```

- Ecriture

```
//Espace de noms: System.IO.IsolatedStorage.  
IsolatedStorageFile store =  
    IsolatedStorageFile.GetUserStoreForApplication();  
    // Créer un nouveau dossier nommé data  
    if (!store.DirectoryExists("data"))  
        store.CreateDirectory("data");  
    // Créer un nouveau fichier nommé donnees.dat  
IsolatedStorageFileStream fs =new IsolatedStorageFileStream(  
@"data\donnees.dat", FileMode.OpenOrCreate, /*Espace de noms  
System.IO*/store); // Ecrire le contenu de la zone de texte dans le  
fichier donnee.dat  
StreamWriter sw = new StreamWriter(fs);  
    sw.WriteLine(textBox1.Text);  
// Fermer le flux  
sw.Close();  
// Libérer les ressources utilisées par le store  
store.Dispose();
```

- Lecture

```
IsolatedStorageFileStream fs=  
    new  
System.IO.IsolatedStorage.IsolatedStorageFileStream  
    ("data\donnees.dat",  
System.IO.FileMode.Open, store)) ;  
StreamReader sr = new StreamReader(fs))  
textBlock1.Text = sr.ReadLine();  
  
    sr.Close();  
// Libérer les ressources utilisées par le store  
store.Dispose();
```

Serialisation

- Deux formats
 - XML
 - JSON

Serialisation XML

- Espace de noms: `System.Xml.Serialization`
- Classe `XmlSerializer`
- La classe à sérialiser doit posséder un constructeur par défaut.

Sérialisation XML

- Une référence de la bibliothèque System.Xml.Serialization doit être ajoutée dans le projet.
- Espace de noms: System.Xml.Serialization;
- Sérialisation

```
IsolatedStorageFile store =  
IsolatedStorageFile.GetUserStoreForApplication();  
IsolatedStorageFileStream fichier= store.CreateFile("data.xml");  
XmlSerializer ser = new XmlSerializer(typeof(List<Departement>));  
ser.Serialize(fichier,deps);  
fichier.Close();  
store.Dispose();
```

- Désérialisation

```
IsolatedStorageFile store =  
IsolatedStorageFile.GetUserStoreForApplication();  
IsolatedStorageFileStream fichier= store.OpenFile("data.xml", FileMode.Open  
);  
XmlSerializer ser = new XmlSerializer(typeof(List<Departement>));  
deps = (List<Departement>)ser.Deserialize(fichier);  
fichier.Close();  
store.Dispose();
```

Sérialisation JSON

- Une référence de la bibliothèque System.ServiceModel.Web doit être ajoutée dans le projet.
- Espace de noms: System.Runtime.Serialization.Json ;
- Sérialisation

```
IsolatedStorageFile store = IsolatedStorageFile.GetUserStoreForApplication();
IsolatedStorageFileStream fichier = store.CreateFile("data.json");
DataContractJsonSerializer ser = new
DataContractJsonSerializer(typeof(List<Departement>));
ser.WriteObject(fichier, deps);
fichier.Close();
store.Dispose();
```

- Désérialisation

```
IsolatedStorageFileStream fichier = store.OpenFile("data.xml",
System.IO.FileMode.Open);
    DataContractJsonSerializer ser = new
DataContractJsonSerializer(typeof(List<Departement>));
deps = (List<Departement>)ser.ReadObject(fichier);
fichier.Close();
store.Dispose();
```