Src : [Serialization (C#) | Microsoft Docs](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/serialization/)

**Sérialisation (C#)**

**Serialization (C#)**

La Sérialisation est le procédé de conversion d'un objet en un flux de bytes pour ranger (stocker) l'objet ou le transmettre à la mémoire ou à une base de donnée ou à un fichier.  
Le but principal est de sauvegarder l'état de l'objet ordonné pour pouvoir le recréer quand c'est nécessaire. L'opération inverse est appelée de-sérialisation.

Serialization is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. The reverse process is called deserialization.

**Comment marche (fonctionne) la sérialisation**

**How serialization works**

Cette illustration montre le procédé global de la sérialisation :

This illustration shows the overall process of serialization:



L’objet est sérialisé dans un flux qui transporte les données. Le flux peut déjà avoir une information à propos du type d’objet, comme (tel que) sa version, sa culture, et son nom d’assemblage. Depuis ce flux l’objet peut être rangé dans une base de données, un fichier, ou la mémoire.

The object is serialized to a stream that carries the data. The stream may also have information about the object’s type, such as its version, culture, and assembly name. From that stream, the object can be stored in a database, a file, or memory.

**Usages de la sérialisation**

**Use for serialization**

La sérialisation permet au développeur de sauvegarder l’état d’un objet et le restituer si besoin, assurer le stockage de donnée autant que l’échange de donnée. Grace à la sérialisation, un développeur peut effectuer des actions comme :

* Envoie d’un objet a une application distante par l’utilisation d’un service web
* Passage d’un objet d’un domaine à un autre
* Passage d’un objet garce à un pare-feu en tant que chaine JSON ou XML
* Maintien de la sécurité ou des informations spécifiques à l’utilisateur entre application

Serialization allows the developer to save the state of an object and re-create it as needed, providing storage of objects as well as data exchange. Through serialization, a developer can perform actions such as:

* Sending the object to a remote application by using a web service
* Passing an object from one domain to another
* Passing an object through a firewall as a JSON or XML string
* Maintaining security or user-specific information across applications

**Sérialisation en JSON**

**JSON serialization**

L’espace de nom (namespace) “System.TextJson” contient des classes pour la sérialisaiton et la de-sérialisation “JavaScript Object Notation” (JSON).

JSON est une norme (Open - Ouverture) standard qui est communément utilisé pour partager des données à travers le web.

La sérialisation JSON Sérialise les propriétés publique d’un objet dans une chaine (de caractère (du texte)), un tableau de byte ou un flux conforme aux spécifications RFC 8259 JSON. Pour contrôler le chemin sérialisé JsonSerializer ou de-sérialiser une instance de classe :

* Utiliser un objet JsonSerializerOptions
* Appliquer les attributs du namespace System.text.Json.Serialization aux classes ou aux propriétées
* Implémenter des conversions personnalisé

The System.text.Json namespace contains classes for JavaScript Object Notation (JSON) serialization and deserialization. JSON is an open standard that is commonly used for sharing data across the web.

JSON serialization serializes the public properties of an object into a string, byte array, or stream that conforms to the RFC 8259 JSON specification. To control the way JsonSerializer serializes or deserializes an instance of the class:

* Use a JsonSerializerOptions object
* Apply attributes from the System.Text.Json.Serialization namespace to classes or properties
* Implement custom converters

**La serialissation Binaire et XML**

**Binary and XML serialization**

L’espace de nom System.Runtime.Serialization contient des classes pour la sérialisation et la de-sérialisation binaire et XML.

La sérialisation binaire utilise l’encodage binaire pour produire une sérialisation compacte pour l’utiliser comme stockage ou flux réseau basé sur le socket. Dans la sérialisation binaire, tous les membres, même les membres qui sont en lecture seul (read-only), sont sérialisé et les performances sont améliorées.

* Attention

La sérialisation binaire peut être dangereuse.

La sérialisation XML sérialise les champs publique et les propriétés d’un objet, ou les paramètres et les valeurs de retour des méthodes, dans un flux XML qui est conforme aux spécifications du document ‘‘XML Schema Definition langage’’ (XSD).

La sérialisation XML aboutis à des classes fortement typé avec des propriétés publique et des champs qui sont convertie en XML. System.Xml.Serialization contient des classes pour la sérialisation et la de-sérialisation XML. Vous appliquerez des attribues aux classe et aux classes membre pour controller la façons dont XmlSerializer sérialisé ou de-sérialisé une instance de classe.

The System.Runtime.Serialization namespace contains classes for binary and XML serialization and deserialization. Binary serialization uses binary encoding to produce compact serialization, all members,even members that are read-only, are serialized, and performance is enhanced.

* Warning

Binary serialization can be dangerous.

XML serialization serializes the public fields and properties of an object, or the parameters and return values of methods, into an XML stream that conforms to a specific XML Schema definition language (XSD) document. XML serialization results in strongly typed classes with public properties and fields that are converted to XML. System.Xml.Serialization contains classes for serializing and deserializing XML. You apply attributes to classes and class members to control the way the XmlSerializer serializes or deserializes an instance of the class.

**Construire un objet sérialisable**

**Making an object serializable**

Pour la sérialisation binaire ou XML, vous avez besoin :

* L’objet à sérialiser
* Un flux pour contenir les objets sérialisé
* Une instance de System.Runtime.Serialization.Formatter

Appliquez l’attribut SerializableAttribute à un type pour indiquer que les instances d’un type peuvent être sérialiser. Une exception est levé (jeté) si vous tenté de sérialiser mais que le type n’as pas l’attribut SerializableAttribute.

Pour empêcher un champ d’être sérialisé, appliquez l’attribut NonSerializedAttribute. Si un champ d’un type sérialisable contient un pointeur, une poignée (handle), ou quel qu’autre structures de données spécifique a un environnement particulier, et les champs ne pouvant être significativement reconstitué dans un environnement diffèrent, alors vous pourrai vouloir le rendre non sérialisable.

Si une classe sérialisé contient des références d’objets d’autres classes qui sont marqué SerialisableAttribute, ces objets seront aussi sérialisé.

For binary or XML serialization, you need :

* The object to be serialized
* A stream to contain the serialized object
* A System.Runtime.Serialization.Formatter instance

Apply the SerializableAttributes attribute to a type to indicate that instances of the type can be serialized. An exception is thrown if you attempt to serialize but the type doesn’t have the SerializableAttribute attribute.

To prevent a field from being serialized, apply the NonSerializedAttribute attribute. If a field of a serializable type contains a pointer, a handle, or some other data structure that is specific to a particular environment and the field cannot be meaningfully reconstituted in a different environment, then you may want to make it nonserializable.

If a serialized class contains references to objects of other classes that are marked SerializableAttribute, those objects will also be serialized.

**Sérialisation basic et personalis**

**Basic and custom serialization**

La sérialisation Binaire et XML peut être effectué de deux façon, basic et personnalisé.

La sérialisation basic utilise .NET pour sérialisé l’objet automatiquement. La seule obligation est que l’attribut SerializableAttribute soit appliqué appliqué à la classe. Le NonSerializedAttribute peut être utilisé pour empêcher des champs spécifiques d’être sérialisé.

Quand vous utilisez la sérialisation basic, le versionning d’objets peut créer des problèmes. Vous voudrai utiliser la sérialisation personnalisé quand la question du versionning est importante. La sérialisation basic est la plus simple pour effectuer des sérialisations, mais il ne réalise pas beaucoup de control sur le processus.

Dans la sérialisation personnalisé, vous pouvez spécifier exactement quel objet va être sérialisé et comment il va l’être. La classe doit être marqué QSerializableAttribute et implémenté l’interface ISerializable . Si vous voulez que votre objet soit également de-sérialisé de manière personnalisé, utilisez un constructeur personnalisé.

Binary and Xml serialization can be performed in two ways, basic and custom.

Basic serialization uses .NET to automatically serializ the object. The only requirement is that the class has the SerializableAttribute attribute applies. The nonSerializedAttribute can be used to keep specific fields from being serialized.

When you use basic serialization, the versioning of objects may create problems. You would use custom serialization when versioning issues are important. Basic serialization is the easiest way to perform serialization, but it does not provide much control over the process.

In custom serialization, you can specify exactly which objects will be serialized and how it will be done. The class must be marked SerializableAttribute and implement the ISerializable interface.

If you want your object to be deserialized in a custom manner as weel, use a custom constructor.

**Le design de serialization**

**Designer serialization**

Le design de sérialisation est une forme spéciale de sérialisation qui implique le type de persistance d’objet associé aux outils de développement. Le design de sérialisation est le processus de conversion d’un graphique d’objet en un fichier source qui peut être utilisé plus tard pour récupérer le graphique de l’objet. Un fichier source peut contenir du code, des balises, ou même des informations de table SQL.

Designer serialization is a special form of serialization that involves the kind of object persistence associated with development tools. Designer serialization is the process of converting an object graph into a source file that can later be used to recover the object graph. A source file can contain code, markup, or even SQL table information.