

Sequence Configuration File Documentation :

English version :	2
Preamble :	2
Definition :	2
Usage :	2
Sequence List Element :	2
Sequence Element :	3
Marker Element :	3
Tips :	3
French version :	4
Préambule :	4
Définition :	4
Utilisation :	4
L'élément "Sequence List" :	5
L'élément "Sequence" :	5
L'élément "Marker" :	5
Astuce :	6

English version :

Preamble :

First, if you have done some tweak to the original file and you want to check if you didn't do any mistakes, without reading all the following explanations, know that you can.

Just head to : <https://www.freeformatter.com/xml-validator-xsd.html>

And either copy-paste your file on the XML section or upload it, then do the same thing for the XSD file called "SequencesDefinition.xsd", that is located under the root folder of the project.

Basically, this XSD file is a set of rules for this specific custom XML format. And if there is any mistake on the XML file, the website will clearly warn you about what it is and where it is.

Also, if you are really new to XML format, you can open the XML file "ConfigSequenceExample.xml", that you will find next to this documentation file. It is a concrete example of what you can do. You can open it on your browser, it will display nicely, but it won't show the notes. For this, use any text-editing program.

Definition :

This XML file is a configuration file that is meant to be used with the LSL Kinect program. It defines a sequence of action for this program. Since it's written on XML, it follows all the XML classic conventions and it's extensible.

If you want to add infos to this file, you can, but be sure to update the XSD file that go along with, and this documentation.

Usage :

For this file to be detected by the program, it needs to be located on the same folder than the .exe file and be named "SequenceConfig.xml".

This file is really basic, it's only compound with three elements. These three elements will be automatically transform into object by the program, and then use by it. So, any mistakes will cause the program to fail to load the instructions contained in these objects.

All the elements tag are case-sensitives.

The elements order at the same level doesn't matter.

Sequence List Element :

This is the root element, the one that need to contains all the others, it has to be called "sequenceList".

- ❖ This element needs to contain at least one "sequence" element.

Sequence Element :

The sequence element, has to be called “sequence”, its purpose is to define a list of actions for the program to interpret.

- ❖ It has an element “name”, that will be displayed on the program, for the user to recognize and select it.
- ❖ And it has a list of “marker” elements, that cannot be empty.

Marker Element :

The marker element, has to be called “marker”, this is the one that truly defines the action the program will execute. Every action defined here will cause a marker, that describes the action, to be sent on the LSL broadcast.

- ❖ It has an attribute called “affectCSV”, that can only be “true” or “false”. It sets if the action should be executed on the CSV too. (Only for the start and stop action). By default it's set to “false”. Notice that, even if the actions don't affect the CSV flow, they will be noted down on the CSV marker file anyway, as marker.
- ❖ It has a child element called “content”. This is the action's description. This description will be sent by the program into a LSL marker when the corresponding action will start. It is also displayed on the program UI when you select the sequence it belongs to.
- ❖ It has a child element called “type”. This is the most important one, it determines which action the program has to do when it finds this marker. You can choose between three action types :
 - Start : Start the LSL broadcast (and the CSV recording if requested).
 - Stop : Stop the LSL broadcast (and the CSV recording if requested).
 - Message : Send a marker containing the “content” element on the LSL stream. It is only useful for the user that will manipulate the XDF output later on.

Even if, there are two action-types called “Start” and “Stop”, a sequence can contain as many “start” and “stop” as you need.

Tips :

If you want to add notes within the XML file, you can add a comment section using these balises :

<!-- This is a comment -->

French version :

Préambule :

Tout d'abord, si vous avez fait des changements au fichier d'origine et que vous voulez vérifier que vous n'avez pas fait d'erreur, sans lire les explications ci-après. Sachez que vous pouvez.

Rendez-vous sur : <https://www.freeformatter.com/xml-validator-xsd.html>

Et soit, copiez-collez le contenu du fichier XML dans la section qui correspond, soit uploadez le fichier. Puis faite la même chose pour le fichier XSD appelé "SequencesDefinition.xsd", qui se trouve à l'adresse dans le dossier racine du projet.

Pour faire simple, ce fichier XSD contient les règles spécifiques au format de ce fichier XML. Et si le fichier XML contient des erreurs, le site web va vous avertir et vous désigner leur emplacement.

De plus, si vous êtes vraiment nouveau au format XML, vous pouvez ouvrir le fichier XML "ConfigSequenceExample.xml", que vous trouverez à côté de ce fichier de documentation. C'est un exemple concret de ce que vous pouvez faire.

Vous pouvez l'ouvrir sur votre navigateur, il s'affichera bien, mais il ne montrera pas les notes. Pour cela, utilisez n'importe quel programme d'édition de texte.

Définition :

Ce fichier XML est un fichier de configuration destiné à être utilisé avec le programme LSL Kinect. Il définit une séquence d'actions pour ce programme. Puisqu'il est écrit en XML, il suit toutes les conventions classiques de XML et il est extensible.

Si vous souhaitez ajouter des informations à ce fichier, vous pouvez, mais assurez-vous de mettre à jour le fichier XSD qui va avec, ainsi que cette documentation.

Utilisation :

Pour que ce fichier soit détecté par le programme, il doit se trouver dans le même dossier que le fichier .exe et être nommé «SequenceConfig.xml».

Ce fichier est vraiment basique, il n'est composé que de trois éléments. Ces trois éléments seront automatiquement transformés en objet par le programme, puis utilisés par celui-ci. Ainsi, toute erreur entraînera l'échec du programme, il ne pourra donc pas charger les instructions contenues dans ces objets.

Tous les éléments sont sensibles à la casse.

L'ordre des éléments au même niveau n'a pas d'importance.

L'élément "Sequence List" :

Il s'agit de l'élément racine, celui qui doit contenir tous les autres, il doit être appelé «sequenceList».

- ❖ Cet élément doit contenir au moins un élément "sequence".

L'élément "Sequence" :

L'élément séquence, doit être appelé «sequence» et a pour but de définir une liste d'actions que le programme doit interpréter.

- ❖ Il possède un élément "name", qui sera affiché sur le programme, pour que l'utilisateur le reconnaisse et le sélectionne.
- ❖ Et il a une liste d'éléments "marker", d'au moins un élément

L'élément "Marker" :

L'élément marqueur, doit être appelé "marker", c'est celui qui définit vraiment l'action que le programme va exécuter. Chaque action définie ici entraînera l'envoi d'un marqueur, qui décrit l'action, sur la diffusion LSL.

- ❖ Il possède un attribut appelé «affectCSV», qui ne peut être que «true» ou «false». Il définit si l'action doit également être exécutée sur le CSV. (Uniquement pour l'action de démarrage et d'arrêt). Par défaut, il est défini sur "false". Notez que, même si les actions n'affectent pas le flux CSV, elles seront de toute façon notées dans le fichier de marqueur CSV, en tant que marqueur.
- ❖ Il a un élément enfant appelé «contenu». Ceci est la description de l'action. Cette description sera envoyée par le programme dans un marqueur LSL lorsque l'action correspondante commencera. Il est également affiché sur l'interface utilisateur du programme lorsque vous sélectionnez la séquence à laquelle il appartient.
- ❖ Il a un élément enfant appelé "type". C'est le plus important, il détermine quelle action le programme doit faire quand il trouve ce marqueur. Vous pouvez choisir entre trois types d'actions:
 - Start : Démarre la diffusion LSL (et l'enregistrement du CSV si demandé).
 - Stop : Arrête la diffusion LSL (et l'enregistrement du CSV si demandé).
 - Message : Envoie un marqueur contenant l'élément «content» sur le flux LSL. Il n'est utile que pour l'utilisateur qui manipulera le fichier de sortie XDF ultérieurement.

Même s'il existe deux types d'action appelés «Start» et «Stop» , une séquence peut contenir autant de «démarrage» et d'«arrêt» que vous le souhaitez.

Astuce :

Si vous souhaitez ajouter des notes à l'intérieur du fichier XML, vous pouvez ajouter une section de commentaire en utilisant ces balises :

```
<!-- Ceci est un commentaire -->
```