XStruct – User Manual

1. About XStruct

XStruct is a program that extracts XML Schemata from given XML data. It can construct concise, complete and understandable schemata from either single XML files or from collections of XML data. XStruct was implemented by Jan Hegewald in the context of a student research project.

XStruct may be parameterized to produce a schema according to the user's needs.

2. Installation

XStruct consists of a single Java jar file named `XStruct.jar`. No installation is necessary except for copying the file to a location on your PC. It can be executed by typing

```
java -jar XStruct.jar
```

on the command-line in the directory where the jar-file is located. When omitting further parameters, XStruct will display a short description of the usage explaining XStruct's syntax and all options.

3. Usage

This section describes how to use XStruct.

3.1. Memory constraints

In general, XStruct is very efficient regarding memory requirements. Nevertheless, as you use XStruct to extract schemata from large XML data, the default heap space size allocated by the Java virtual machine might be not sufficient. If this happens, you get an error similar to the following:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
```

In order to tell the Java virtual machine to allocate enough memory, you have to specify two parameters when starting Java: `-Xmx` for the maximal size of heap space and `-Xms` for the minimal size to allocate by Java. E.g. you will need about 850 MB of heap space when processing 1 GB XML data.

Sample values for initially 20 MB of heap space which Java can extend to a maximum of 200 MB when needed are:

```
java -Xms20m -Xmx200m -jar XStruct.jar
```

For more detailed information on this topic review the documentation of your Java installation.

3.2. Parameters

XStruct may be used in two ways. Either you may specify a configuration file that contains all the settings or you may specify them directly on the command-line.

- For specifying the options on the command-line, XStruct's call syntax is as shown below:

```
XStruct [options] [-output schemafile.xsd] inputfile1.xml inputfile2.xml ...
```

Here, options are some of the following:

Option	Meaning	Default
`-Threshold`	A numeric value for the threshold used in the inferring of element content models. This has to be a positive integer number.	`2`
`-MaxSavedValues`	A numeric positive or zero value that determines how many different values for an element or attribute are stored. Larger numbers increase the precision in data type detection but also require more memory during the processing.	`15`
`-MaxValuesEnumeration`	A numeric positive value that determines how many different values may be used for an enumeration before XStruct discards the enumeration and instead sets a primitive datatype. E.g. if `MaxValuesEnumeration` is 10, XStruct will print every element as an enumeration that contains no more than 10 different values in its occurrences.	`10`
`-LimitUnboundness`	A numeric positive value that determines how often an object has to appear to set its `maxOccurs`-value to `unbound`.	`10`
`-SchemaNamespace`	A string that specifies which namespace prefix XStruct should use when constructing the XML Schema.	`xsd`

Table 1: Possible Options

An output file may be specified but does not have to. If you omit this, XStruct will print the generated schema to the standard output. Important to know is that the input files always come last.

A possible valid call would be:

```
java -Xmx200m -Xms20m -jar XStruct.jar -SchemaNamespace xs -LimitUnboundness
20 -output result.xsd data1.xml data2.xml
```

- When specifying options in a separate configuration file, XStruct's syntax is as follows:

```
XStruct -conf configurationfile [-output schemafile.xsd] inputfile1.xml
inputfile2.xml ...
```

`configurationfile` has to be the path to a file containing name-value-pairs. The name-value-pairs have the same meaning as described in Table 1: Possible Options. A config-file that uses the same options as the example for command-line above, would look like this:

config.properties:

```
SchemaNamespace=xs
LimitUnboundness=20
```

XStruct then could be called using this configuration file as follows:

```
java -Xmx200m -Xms20m -jar XStruct.jar -conf config.properties -output
result.xsd data1.xml data2.xml
```