# Nordic to QuakeML converter

This application, called nor2qml, is a conversion tool between the Nordic format and the QuakeML (XML representation) format. The tool is preconfigured to read s-files (Nordic format) and producing earthquake event data (event parameters) in the QuakeML format. By configuring arguments, the tool can also convert QuakeML data into a collection of s-files from both an XML file and a web-service. The application is runnable within the SEISAN environment or directly through the java environment (jar files).

The following description of the nor2qml converter is divided into four sections:

1. How to run the application
2. Configure IDs for QuakeML output
3. Configure arguments for conversion from QuakeML to nordic
4. Arguments

The two first sections describe the conversion from the Nordic format to QuakeML. The third section gives an explain on how to enable conversion from QuakeML to the Nordic format. The fourth section gives an overview of all the possible arguments that can be used within this application along with default values for each argument.

# 1. How to run the application

Make sure that you have java installed. Check by typing “java -version” into the terminal.

To run the application, type the following command into the terminal:

* For SEISAN environment:

*“nor2qml --input=”<path to file or folder>” --output=”<path to file>”*

* For Java environment:

*“java -jar nor2qml.jar --input=”<path to file or folder>” --output=”<path to file>”*

For running the application within the Java environment, specify the entire path to the nor2qml.jar after the “-jar” argument.

## Input

The application has a built-in file scanner, which is designed to identifying s-files within in SEISAN catalog structures. Hence, a folder will only be scanned if the folder name follows the convention of a catalog name with underscore (“\_”) at the end, for example the catalog name “NNSN\_”. The file scanner will also search through folders with a folder name of numbers, which is true for the catalog’s subsequent folders of year and month. Examples of valid input for folders are:

* Folder (whole catalog): “<path to seisan>/SEISMO/REA/NNSN\_”
* Folder (specific year): “<path to seisan>/SEISMO/REA/NNSN\_/2014”
* Folder (specific month): “<path to seisan>/SEISMO/REA/NNSN\_/2014/09”

During file scan, only files that has the extension starting with “S” will be read. However, if the given input path is pointing at a single file, the application can read any file regardless of extension or filename. Hence, following examples of file input can be used:

* File (particular file): “<path to seisan>/SEISMO/REA/NNSN\_/2014/09/01-2050-05L.S201409"
* File (particular file): “<path to any folder>/testfile.out”

## Output

The output argument has a default value of the path where the application (jar file) is located. If you are running the converter through the seisan structure it is recommended to specify another path for the output. The ouput value should the complete path with a filename and extension at the end of the path. If converting from Nordic to QuakeML, please give the “xml” extension. If converting from QuakeML, you are freer to choose extension, but as a guideline you can use the “.out” extension.

## Example for running the application with input and output specified

* For SEISAN environment:

*nor2qml --input=”/home/SEISMO/REA/NNSN\_/2014/09” -output=”/home/my-name/qml201409.xml”*

* For Java environment:

*java -jar nor2qml.jar --input=”/home/SEISMO/REA/NNSN\_/2014/09” -output=”/home/my-name/qml201409.xml”*

# 2. Configure ID generation

Within QuakeML, each entity has a public ID for identifying each resource given. This may for instance be a particular event, origin, magnitude, or amplitude. The resource IDs is crucial for linking these entities. The application has a build-in ID generator that follows the QuakeML standard of resource identifiers. By default, these identifiers are built by some predefined values and attributes given by the event itself. The attributes taken from the events are fixed, but two of the predefined values can be overwritten by providing arguments. Since the application provide default values, the conversion can take place without specifying these arguments, but it is recommended that you at least give a value for the authority ID as this will link the resource to your institution. Before explaining which arguments to use, a brief overview of the resource identifiers by QuakeML is given.

As written in the QuakeML documentation1, the identifiers should follow this generic form:

*[smi|quakeml]:<authority-id>/<resource-id>[#<local-id>]*

More explicitly the id follow this schema pattern:

*( smi | quakeml ) : [ \w\d ] [ \w\d \−\.\∗\(\) ˜ ’ ] { 2 , } / [ \w\d \−\.\∗\(\)˜ ’ ] [ \w\d \−\.\∗\(\) \+\?˜’=,;#/&amp ; ] ∗*

The two first part of this identifier is configurable in the application by the arguments *prefix* and *agency*.

## Prefix

The prefix argument takes a sting value that should be either “smi” or “quakeml”. The default value is “smi” and if you want to change this, simply attach --prefix=”quakeml” when running the application. “smi” stands for seismological meta-information.

## Agency

Similarly, the authority-id can be configured by attaching --agency=”<your authority-id>”, which should be at least 3 characters with the first charcter as alphanumeric, while subsequent characters can be alphanum or from the following list: -, \_, ., ~ , \*, ’, (, ).  
  
The agency-id should follow this form:  
<top-level domain>.<organisation/institution>[.<sub-unit of organisation>]

An example can be “no.uib.nnsn”, where “no” (Norway) is the top-level country domain, “uib” (University of Bergen) is the organisation and “nnsn” (Norwegian National Seismic Network) is the sub-unit of the organisation.

## Example with ID generation

Running the application for an s-file within the NNSN catalog could then look like this:

* For SEISAN environment:

*nor2qml --input=”/SEISMO/REA/NNSN\_/2014/09” -output=”/home/my-name/qml201409.xml” --prefex=”smi” --agency=”no.uib.nnsn”*

* For Java environment:

*java -jar nor2qml.jar --input=”/SEISMO/REA/NNSN\_/2014/09” -output=”/home/my-name/qml201409.xml” --prefex=”smi” --agency=”no.uib.nnsn”*

# 3. Configure to convert from QuakeML to Nordic

Conversion from QuakeML to the Nordic format can be achieved from two different sources, an XML file or a web-service.

## 3.1 QuakeML file to Nordic

To convert a QuakeML file to the Nordic format, three arguments is required: *convert*, *input and output* arguments. For SEISAN environment, the convert argument is already predefined and can thus be omitted. Description about the input and output parameters are given in the section above on how to run the application.

For Java environment, the convert argument is by default set to “q” (meaning QuakeML), specifying the output format. Similarity “s” can be given to point to s-file output in the Nordic format. Thus, in order to convert to Nordic, attach --convert=”s” to the command. The order of arguments does not matter.

The full command would then look like:

* For SEISAN environment:

*qml2nor --input=”<path to file>” --output=”<path to file>”*

* For Java environment:

*java -jar nor2qml.jar --convert=”s” --input=”<path to file>” --output=”<path to file>”*

## 3.2 QuakeML web-service to Nordic

Configuring to read from a QuakeML web-service requires four arguments: *output*, *convert*, *source* and *url*. Both output and convert are set in a similar fashion as for converting a QuakeML file (see previous section). The source parameter is telling the application that it should fetch data from either a file or an external web-service. The default source value is file. To fetch data from a web-service, attach –source=”ws” to the command. As a web-service is an external source, an internet connection is required to make use of this command. In addition to source, an URL to the web-service is also required and should be given in with the URL parameter, by attaching –url="<path to web-service>".

The full command would then look like:

* For SEISAN environment:
  + qml2nor--output="sfiles.out" --convert="s" --source="ws" --url="<path to web-service>"
* For Java environment:
  + java -jar nor2qml.jar --output="sfiles.out" --convert="s" --source="ws" --url="<path to web-service>"

# 4. Arguments

### General Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Value type** | **Default value** | **Description** |
| --convert | “q” or “s” (type String) | “q” | Specify to which format the files should be converted to. “q” means convert from s-files to QuakeML, while “s” means convert QuakeML to s-files. |
| --input | Path (type String). The value must be enclosed with “”. | Current path | Select where the input files are. If no path is specified, the program will search the current folder for files to be converted. |
| --output | Path (type String). The value must be enclosed with “”. | Current path | Select where the converted file should be placed. If no path is specified, the program will create a single file in the current folder with the filename and extension to either “quakeml.xml” or “sfile.out” based on the conversion type. |
| --version | “nordic”, “nordic2”, “1.2” or “2.0” | “1.2” or “nordic” | Specifying the version of the output format. Can be both version Nordic and QuakeML format. |
| --eventtype | Type String. Supported events from the enumeration list of the QuakeML EventType in lower case text. | “earthquake” | Specifies the default event type of the Event element if no type is given in the s-file. To leave out event type from QuakeML output, insert a text string that does not match with the enumeration list. |
| --eventcertainty | Type String. Can be either of these: “known”, “suspected” | “suspected” | Specifies the default event certainty of the Event element if no certainty is given in the s-file. To leave out event certainty from QuakeML output, insert a text string that does not match with “known” or “suspected”. |

### QuakeML specific arguments

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Value type** | **Default value** | **Description** |
| --prefix | String that should be “smi” or “quakeml” | “smi” | Specifying the prefix of generation of IDs |
| --agency | String | “authorityid” | Specifying the agency for generation of IDs (e.g. “no.nnsn”) |
| --source | String that should be “file” or “ws” | “file” | Specify if the source is a file or a web-service (ws) |
| --url | String (mandatory if source is ws). The value must be enclosed with “”. |  | Specifying the url if the source is a web-service |

# References

[1] <https://quake.ethz.ch/quakeml/docs/REC?action=AttachFile&do=view&target=QuakeML-BED-20130214b.pdf>