

## Ninestars QA Suite

The Ninestars QA Suite consists of two tools for checking the newspaper scans for The State and University Library in Aarhus, Denmark.

One tool is designed to be run on a single JPEG 2000 file, and will report any deviations from the specification of the file. The other tool is designed to be run on a directory structure containing the output of an entire batch, including data and metadata, and will report any deviations from the specifications.

### What is validated

The following things are validated:

- All jp2 files are validated to conform to the specifications, as specified in Appendix 2B
- All md5sums are checked to be correct, as specified in Appendix 2F
- All directories are checked to have the correct names and contain the correct files, as specified in Appendix 2F
- All files are checked to have the correct names, as specified in Appendix 2F
- The [AvisID] and [date] are checked to conform to the data in the database
- All XML files are checked to conform to the correct XML Schema as specified in Appendix 2C, 2D, 2E, 2J and 2K
- Automatic check of contents in XML metadata as specified in Appendix 2C, 2D, 2E, 2J and 2K

### Prerequisites

Prerequisites include:

- A linux system. The tool has been tested on [CentOS](#) release 6.4, but most linux versions should work, provided the other prerequisites can be met.
- [Java SE 7](#)
- [Python](#) version 2.7 or above or 3.2 or above
- Access to the SULA provided Postgres database, or a mirror of it.

This list may be extended.

### Running

#### The file tool

Start the file tool with

```
bin/qafile.sh <path-to-file>
```

<path-to-file> is the path to the file to check.

## The batch tool

Start the batch tool with

```
bin/qabatch.sh <path-to-batch> <sql-connection-string>
```

<path-to-batch> is the path to the directory containing the batch to check.  
Example: “/var/spool/batch/B400022028241-RT1”

<sql-connection-string> is the connection string used to connect to the database with the information about the batch. Example: “jdbc:postgresql://dbhost/mfpak?user=mfpak&password=mfpass”

The database contains information about the data in a batch.

## Output

Both tools will output an XML file to stdout, containing the result of the validation (success or failure) and a list of errors, if any.

The schema for the XML file can be found here [Schema](#).

In general, messages will refer to the specification, when reporting an error. For errors regarding file structure, refer to this [list of checks](#) for details.

Examples of output can be found here: [Success](#) and [Failure](#)

If the tool fails unexpectedly, it will exit with an error code larger than zero, and error messages will be printed to stderr.

The tool also produces a log while executing. By default the log will be output to stderr with a level of WARNING. This can be changed by editing the file conf/logback.xml. You can configure logging as described in the [Logback manual](#).