NAHLN-O-MATIC\_AMR

User Guide

This is a simple, single purpose, NAHLN message generator and sender. It uses file folders for all input and output. The only user interface is a progress box. This requires Java version 8 or higher installed and configured to execute runnable .jar archive files.

The *only* thing this program does is read Excel workbooks in the format specified by the AMR Pilot Project, convert each row in each sheet to NAHLN message format and send as a message. The only setup besides populating the source spreadsheets is in tables in the Config.xlsx file.

Package contents:

* NAHLN-O-MATIC\_AMR.jar The program.
* Config.xlsx Almost all configuration and mapping tables in Excel sheets.
* LogConfig.txt Settings for logging that must load before above can be read. Ignore.
* LICENSE.txt Apache 2.0 open source license for this code and included libraries from Apache.

Folders:

* Main NAHLN-O-MATIC folder Where ever you want it.
* InBox In the main folder. Contains spreadsheets to process.
* OutBox In the main folder. Receives processed spreadsheets, messages, and acknowledgments for AA responses.
* ErrorsBox In the main folder. Receives processed spreadsheets, messages, and acknowledgments for AE and AR responses.

Configuration:

The Config.xslx file contains four tabs: One with configuration settings and three with mapping tables. Most of the settings in Config should be apparent. Folder locations for the three main folders can be changed but I don't know why you would. The *Host* is the NAHLN LMS service, either Test or Production. It is set to HL7ResultTest as shipped. *User* and *Password* are your NAHLN Messaging System credentials. *Mode* is D for initial debugging, T for testing with NAHLN, and P once you change over to the production server. Your *LabPIN* and *LabOID* are the same as for regular NAHLN messaging. The *NahlnOMaticOID* needs to be a branch off of your lab OID. I used .500. Since each install of NAHLN-O-MATIC keeps track of its own unique IDs, it is important that this OID be unique. *LogLevel* can be INFO or ERROR. Change to ERROR once it is working to keep your log file smaller. *ProfileID* is the filename of the current NAHLN schema without the extension. And *PollSeconds* is how often the program looks for new files in the InBox folder.

The SNOMED tab contains all the various SNOMED code mappings used for Taxomony, Specimen Type, and Isolate Identification. The first column must match what you put in the spreadsheet ***exactly***. The exception to this is for *Salmonella enterica*. These match the Bacterial Organism Isolated and Salmonella Serotype columns combined. SNOMED is hierarchical so the organism identified names both the species and serotype. The second column is the SNOMED code and the third column is the SNOMED fully specified name.

The LOINC tab works similarly. You shouldn't need to edit this tab because the tests listed in the spreadsheet are already in here as are the panel, culture, and diagnosis codes. ***DO NOT*** modify the columns or headers in the spreadsheet in any way or the results may not map correctly. If the AMR project changes the template, we will publish an updated .jar file.

The Local tab contains the "Local" (to the NAHLN) list of Reasons for Testing. The first column must match what you put in the spreadsheet ***exactly***. If you put a slight variation, simply put that in the first column of a new row and add the code and text that correspond. If you have three people doing the spreadsheets and each words it differently, just do three rows of each. But it must match.

Note: Changes to Config.xlsx take effect the next time you start the program.

Running:

Simply run the NAHLN-O-MATIC\_AMR.jar file—by double clicking if your Java is setup right—and a progress dialog box will appear. Drop completed spreadsheet templates into the InBox folder. The progress dialog will report on its progress. If you have more than one row filled in it may go by too fast to read. Look in the ErrorsBox folder for any AE messages and the corresponding Spreadsheet and message files. At that point you are on your own to use Notepad++ or some other tool to read and diagnose the errors.

Known Bugs/Issues:

The program has not been tested on spreadsheets with some valid and some invalid messages produced. The Excel data file should end up in ErrorsBox if any row on any sheet has an error. The Ack and Message files will be in the appropriate box. However, if an unexpected error (internal error, data corruption, malformed Excel sheet, etc. occurs, the source file moves to the ErrorsBox and it will take some doing to figure out what has been sent by looking in the OutBox.