FOBIA
StanForD 2010 MOM Parser

User manual

JavaScript application description

2020 LUKE

Northern Periphery and Arctic Programme
2014–2020

EUROPEAN UNION

Investing in your future
European Regional Development Fund

StanFord 2010 MOM Parser

User Manual

FOBIA

STANFORD 2010 MOM PARSER

Upload one or more mom files for inspecting or exporting as row data

UPLOAD | ABOUT

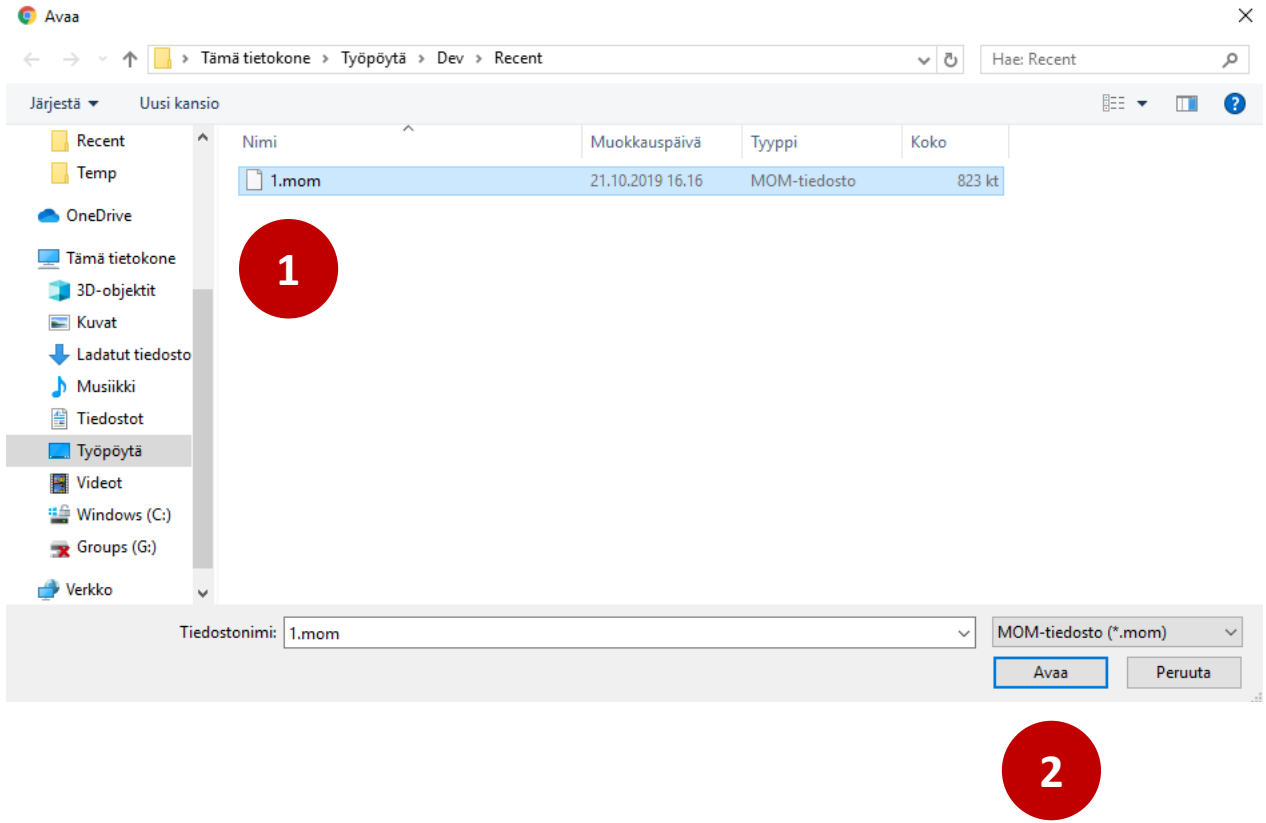
Information on the FOBIA project and processed row data Excel importer

Northern Periphery and Arctic Programme 2014–2020

EUROPEAN UNION

Investing in your future
European Regional Development Fund

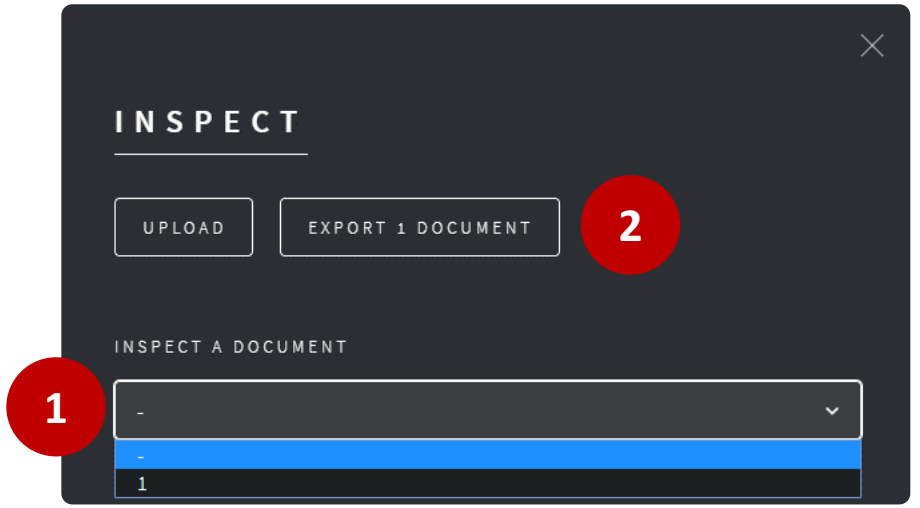
Importing MOM Files



With file selector (1) select one or more MOM files and finally click Open button (2)

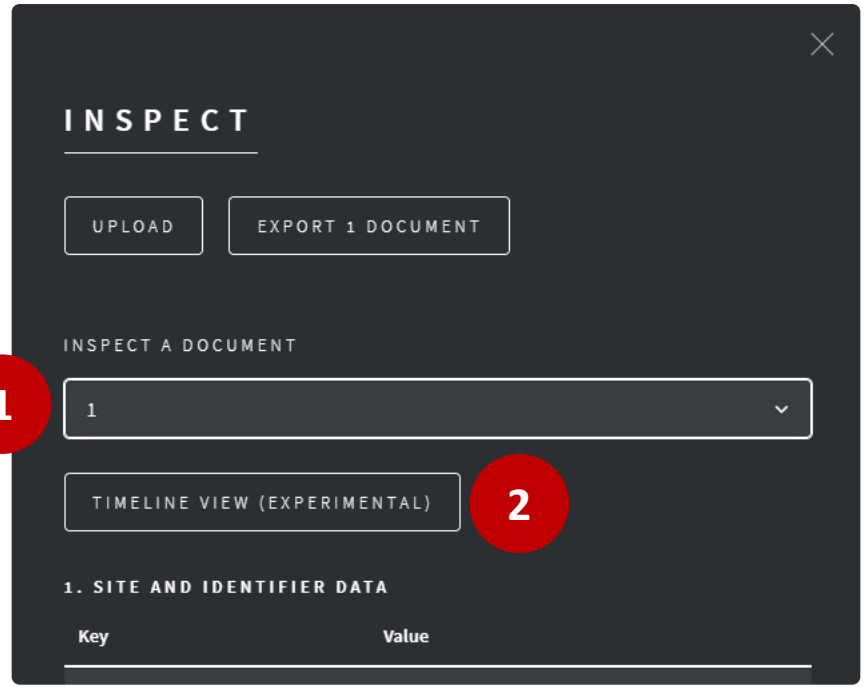
Inspect and Export

After file upload the app is redirected to inspector where you can view (1) individual files or export (2) them to row data



Inspecting a single MOM File

With Inspect a document drop down menu (1) select a file to inspect. Experimental timeline view (2) shows selected time stamps (operator work time, machine work time and machine down time) in horizontal timeline.



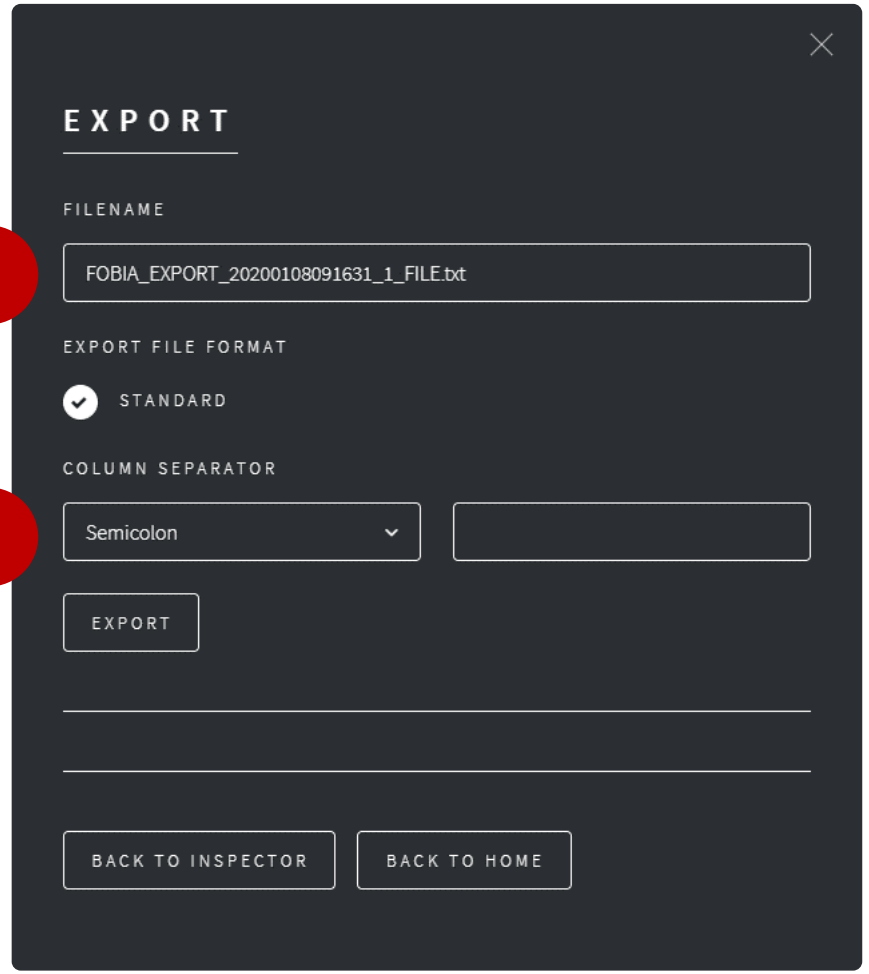
Timeline view:

Exporting MOM Files as row data

Export reads all the imported MOM files and converts them to a single file for further analysis.

In this form you can change the export file name (1) and the column separator (2) in row data.

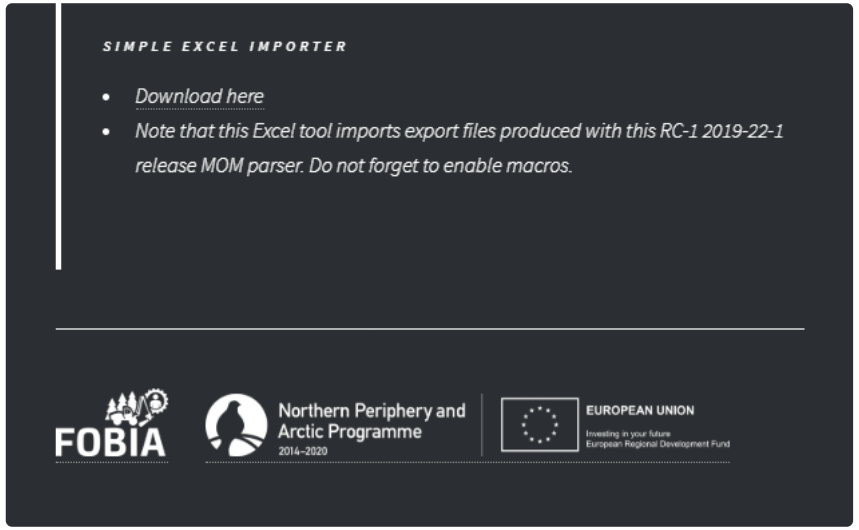
Please note, that the Excel importer provided by the FOBIA project reads successfully only files with semicolon separator.



Simple Excel Importer Download

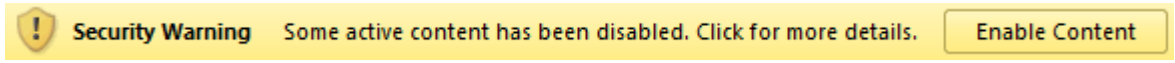
In About page you'll find the
Simple Excel Importer download link.

Download the importer and enable the macros
in it in order to be able to run the application.



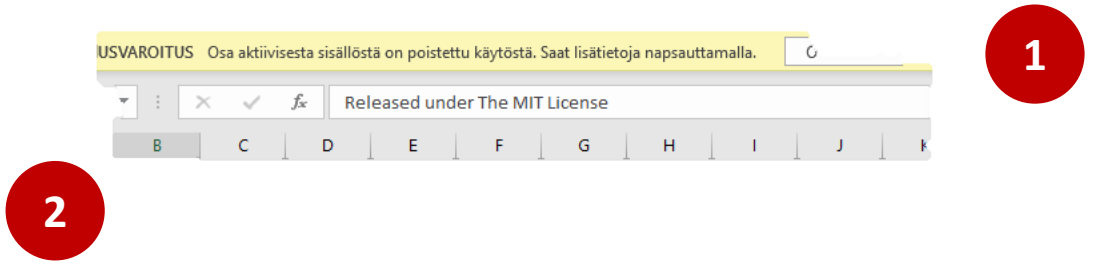
Simple Excel Importer

In order to application to run as intended it is important that you enable the content using the button (1) after the security warning.



Use Import button (2) to open the file selector. Select a file to import and finally click the OK button.

Excel importer formats the row data and calculates some infile specific sum values when applicable.



StanFord 2010 MOM Parser

JavaScript application description

JavaScript application overview 1/2

FOBIA StanForD 2010 MOM parser is a single page web application with some complementary custom JavaScript files. The DOM manipulation is performed using vanilla js without any third party frameworks (Angular etc).

Project file overview:

- index.html
  - Main app page
- app/bodybuilder.js
  - Routines for manipulating index.html DOM
- app/xml2json.js
  - Function for converting XML to JSON for easier manipulation in JavaScript
- app/utility.js
  - Time (incl date) calculation and conversion functions
  - Some auxiliary functions for array manipulation and filetype detection
- app/stanford2010parser.js
  - Functions for StanForD 2010 variable extraction. Detailed function description in appendix A.
- app/fileio.js
  - Functions for taking 1-n files from HTML file input element and returning JSON objects in array. Uses xml2json.js
- app/export.js
  - Functions for creating export file from source data

JavaScript application overview 2/2

Application runtime logic flow goes as follows:

- Clicking Upload button calls the checkInputStatus() function in bodybyilder.js that checks the input status and sets the input button view string from Upload to Inspect if input array has some payload. If the input array is empty the function calls the fileDialog() function in fileio.js
- If files were selected the fileDialogInput() function (fileio.js) is called and the MOM files as JSON passed to myFileReader(file, callback) (fileio.js) call back function being redirect() that finally calls inspect() function in bodybuilder.js
- Export # files button in inspector form calls the exportBuilder() in export.js and passes the payload (uploaded MOM files as JSON array) to it for processing. Finally the exportBuilder() calls the exportForm() function in bodybuilder.js that parses the export dialog accordingly.
- Export button in export dialog calls the download_simple() function in export.js that uses simple blob-anchor paradigm creating upload file object

APPENDIX A – stanford2010parser.js

Functions

- getJsonReturnType(object)
  - Returns the type of a JSON Object AS STRING
- OPERATIONAL_MONITORING_MACHINE_getValue(momAsJSON, momKey)
  - Returns requested object AS JSON OBJECT, if exists, from document-OperationalMonitoring-Machine
- OPERATIONAL_MONITORING_MACHINE_getWorkTimeObjectType(momAsJSON)
  - Returns machine work time object type AS STRING. Possible return values UNDEFINED, INDIVIDUAL, COMBINED
- getOperatorName(momAsJSON, operatorKey)
  - Returns operator Name as STRING
- OPERATIONAL_MONITORING_MACHINE_getOperatorKeys(momAsJSON)
  - Returns operator keys AS ARRAY
- OPERATIONAL_MONITORING_MACHINE_CATEGORY_getValue(momAsJSON)
  - Returns machine category (harvester, forwarder) AS STRING
- getMachineWorktimeTimestamps(momAsJSON)
  - Returns all the timestamps (start and end times) in machine work time objects in milliseconds AS ARRAY
- getMachineWorkTimeStartTime(momAsJSON)
  - Returns start time on site AS DATETIME FORMATTED STRING
- getMachineWorkTimeEndTime(momAsJSON)
  - Returns end time on site AS DATETIME FORMATTED STRING

APPENDIX A – stanford2010parser.js

Functions

- OPERATIONAL_MONITORING_MACHINE_OBJECT_DEFINITION_getLoggingOrganisation(momAsJSON)
  - Returns the type of a JSON Object AS STRING
- OPERATIONAL_MONITORING_MACHINE_OBJECT_DEFINITION_getLoggingForm(momAsJSON)
  - Returns logging form AS STRING
- OPERATIONAL_MONITORING_MACHINE_getMonitoringSettingsReturnType(momAsJSON)
  - Returns monitoring settings (filter times) object type AS STRING
- OPERATIONAL_MONITORING_MACHINE_getMonitoringSettings(momAsJSON)
  - Returns monitoring settings (filter times) object AS JSON OBJECT
- OPERATIONAL_MONITORING_MACHINE_MONITORING_SETTINGS_getMonitoringFilterTimeDown(monitoringSettingsObject)
  - Returns FilterTimeDown from monitoring settings (filter times) object AS STRING
- OPERATIONAL_MONITORING_MACHINE_MONITORING_SETTINGS_getMonitoringFilterTimeRun(monitoringSettingsObject)
  - Returns FilterTimeRun from monitoring sttings (filter times) object AS STRING
- OPERATIONAL_MONITORING_MACHINE_MONITORING_SETTINGS_getMonitoringFilterTimeMinimum(monitoringSettingsObject)
  - Returns FilterTimeMinimum from monitoring settings (filter times) object AS STRING
- getSumOfDrivenDistance(momAsJSON, operatorKey)
  - Returns operator specific sum of machine driven distance from machine work time objects AS INTEGER
- getSumOfFuelConsumption(momAsJSON, operatorKey, filter)
  - Returns operator specific sum of fuel consumption AS INTEGER. Filter values Processing, Terrain travel, Other work, null (returns total fuel consumption)

APPENDIX A – stanford2010parser.js

Functions

- getSumOfEngineTime(momAsJSON, operatorKey)
  - Returns operator specific sum of engine time in hours AS FLOAT
- getSumOfForwarderLoadCount(momAsJSON, operatorKey)
  - Returns operator specific sum of forwarder load count AS INTEGER
- getOperatorWorkTimeInCategory(momAsJSON, categoryString, operatorKey)
  - Returns sum of operator work time (seconds) in requested category AS INTEGER. Request category strings: MachineWorkTime, Meal break, Planning outside, Travelling to work outside machine, Other work outside machine
- getMachineWorkTimeInCategory(momAsJSON, categoryString, operatorKey)
  - Returns operator specific machine work time (seconds) in requested category AS INTEGER. Request category string: Processing, Terrain travel, Other work
- getMachineWorkTimeInSubCategory(momAsJSON, categoryString, subcategoryString, operatorKey)
  - Returns operator specific machine work time in requested machine work time sub category AS STRING. Request category string: Processing, Terrain travel, Other work. Request sub category string: Processing categories SingleTreeProcessing, MultiTreeProcessing, SingleTreeFelling, MultiTreeFelling, Other work, Road travel, Preparing strip roads, Towing other machine, Roadside loading of truck, Unspecified
- getFilteredMachineWorkTimeInCategory(momAsJSON, categoryString, operatorKey, filterValueSeconds)
  - Returns filtered operator specific machine work time (seconds) in requested category AS INTEGER. Request category string: Processing, Terrain travel, Other work
- getMachineDownTime(momAsJSON, operatorKey)
  - Returns total sum of operator specific machine down time (seconds) AS INTEGER
- getFilteredMachineDownTime(momAsJSON, operatorKey, filterValueSeconds)
  - Returns filtered total sum of operator specific machine down time (seconds) AS INTEGER

APPENDIX A – stanford2010parser.js

Functions

- getDownTimeArray(momAsJSON, operatorKey)
  - Returns operator specific down times in objects AS ARRAY
- getMachineDownTimeType(downTimeObject, downTimeObjectType, filter)
  - Returns machine down time type AS STRING. Filters: ROOT, STANDARD_CODE, MANUFACTURER_CODE. Down time object type: INDIVIDUAL, COMBINED
- getSumOfForwardedVolume(momAsJSON, operatorKey)
  - Returns sum of operator specific forwarded volume (m3sob and m3sub) as ARRAY
- getSumOfVolumeOfStems(momAsJSON, operatorKey)
  - Returns sum of operator specific volume of harvested stems as key value pairs in ARRAY
- getSumOfVolumeOfStems_volumeFiltered(momAsJSON, operatorKey, volumeFilter)
  - Returns sum of operator specific filtered volume of harvested stems as key value pairs in ARRAY
- getSumOfVolumeOfStemsRecoded_volumeFiltered(momAsJSON, operatorKey, volumeFilter)
  - Returns recoded (species group) sum of harvested stems filtered volume aggregate as ARRAY
- getSumOfVolumeOfStemsRecoded(momAsJSON, operatorKey)
  - Returns recoded (species group) sum of harvested stems volume aggregate as ARRAY
- getSumOfHarvestedStems(momAsJSON, operatorKey)
  - Returns operator specific sum of harvester stem count as key value pairs in ARRAY
- getSpeciesGroupName(momAsJSON, speciesGroupKey)
  - Returns SpeciesGroupName as STRING

APPENDIX A – stanford2010parser.js

Functions

- getRecodedSpeciesGroupString(speciesGroupString)
  - Returns recoded (English) species group string (name) as STRING
- getSumOfHarvestedStemsRecoded(momAsJSON, operatorKey)
  - Returns recoded (species group) sum of harvested stems aggregate as ARRAY
- getSpeciesGroupKeys(momAsJSON)
  - Returns unique species group keys as ARRAY

Additionally the stanford2010parser.js includes number of functions not used in the final product. Functions, if not stated otherwise in comments, are functional.