**MCM Importer for Kintecus v0.4.1**

7/2/2015

By Fred Winiberg

Script written in VBA to convert the MCM v3.2 Facsimile output into Kintecus input format. Currently only works for the Excel based version of Kintecus. This is a vague manual that explains the process of the code currently (v0.4.1). Drastically needs updating with some sort of guide on how to use it. But for now:

1. Open a blank MCM workbook.
2. Import the MCM output (Facsimile format) into a worksheet, all in column A, without delimiting the file. Rename the worksheet “MCM”. The script will not work properly if the MCM worksheet is not present and the model file is not all in column A.
3. Hit the “Import MCM model from Spreadsheet” button. This will overwrite the model and species spreadsheets in the open kintecus file.
4. Make sure that the ‘parm’ worksheet has the Ea units set to ‘KELVIN’ and the conc units are set to ‘MOLECULES/CC’. The MCM relies on these two settings.
5. Run the model as normal!

The script is not perfect and there are certain reactions that the MCM does not assign products (usually resulting in O2 or H2O), and hence these will need to be added to get Kintecus to work. However, when the model is run, Kintecus will identify these for you. Hopefully I will figure out a way of sorting this one day…

The file ‘MCM\_Constants.txt’ must be placed in the Kintecus directory for the script to work correctly!

**How it works:**

**Importer:**

* User must fist import the Facsimile format file into excel without any delimiters into a spreadsheet named "MCM". All data are in Column A only. Do not include
* Script is essentially a series of nested IF statements inside a for loop that looks for specific strings in the Facsimile file, reading the file line by line.
  + VARIABLE - lists the species used in the reaction. Used here to create the "Species" spreadsheet in Kintecus.
  + RO2 - Peroxy radicals used in the model. Used by the Species spreadsheet to calculate the total RO2 used in the model.
  + % - Percentage sign determines a reaction. Further nested IF statements look for specific characters and reaction names to determine input style for Kintecus "model" spreadsheet. More on these below.
* The code is able to look for multiple lines of code, until it finds a Facsimile termination character, ";". This can cause hiccups if some lines have wrapped around!
* Once a reaction line is found (% character) the script splits the line into a rate expression and reaction term, which are colon delimited.
* The reaction term has the semi-colon and end white space removed, followed by replacing the Facimile "=" with the Kintecus "==>". The rate string is held in memory until the end of the loop.
* Some reaction rate coefficients are special and require identifying before the main loop iterates. This is done by looking for the "\*(" string as the special reactions from the MCM all start with this. The contents of the brackets, i.e. "\*(...)" are checked against a hard coded case structure, which determines their equivalent complex rate expression beginning with "K..." (See below). Once this is assigned, the "\*(...)" is substituted for the complex rate expression and the main loop is started.
* Then, the script looks for unique identifiers in the rate expression to determine the rate expression format (unlike facsimile, Kintecus requires a specific rate expression format). These parameters are mostly separated by "\*" and hence the script separates the rate expression by "\*" (there may have to be a different way to do this eventually), and goes through these delimited values in order (causes problems with some special rate expressions and multiple [M] species).
  + Numeric - i.e. a pre-exponential factor or branching ratio. These values are multiplied together and put in column A. Multiplications are done in Excel so that the user can identify the calculation has been correct. Good for debugging.
  + "EXP" - The script identifies this as an "EXP(Ea/TEMP)" type expression and extracts the Ea (in Kelvin), multiplied by -1 here, as Kintecus computes "(-Ea/Temp)" explicitly.
  + "TEMP@" - This is used in the modified Arrhenius type expressions, T^n, which is a standard input for Kintecus. The number after the "@" is stored in memory.
  + "K..." - All complex rate expressions start with a capital K and so this general term has been used to identify these. The script then examines the whole name and searches for the match in a constants text file (MCM\_Constants.txt) contained within Kintecus directory. These are organised in such a way that the code identifies the corresponding Kintecus special reaction name (MCM1, MCM2, MCM3 etc.) and parameters required (k\_inf, n, k\_0, m etc.) and formats them in the general way:

"...+M[MCM1; 4.5E-31; 3.9; 0.0000000000013; 0.7; 0.525;]"

* + "TEMP/300" or "TEMP/298" - These are identified and assigned to MCM4 kinetic form.
  + Third body enhancers are collected up and added to the "+M[...]" expression at the end of the loop iteration. All except "RO2", which is added directly to the reaction string.
* The final part writes the model sheet with the data stored in memory up to this point. The loop repeats for the next line of the Facsimile file.
* If a rate coefficient is unidentified then the importer will output these unidenitified reactions with their corresponding Facsimile formated rate coefficient at the end of the model spreadsheet after all other data has been sorted.