# Steps to derive JS objects from csv for the WHAF decision support module

1. Original table needs to be a csv, and must be based on the csv template under the ‘scripts’ directory in the WHAF app.
2. Note: values in column A, which includes the names of the indices, must not be changed. The index names need to be exactly identical to the way they appear in the app’s inside (‘indexdescNewJson’ object). Rows for each index may be added.
3. All other values can be changed and updated, including the system benefit names (more columns may be added); association between component and system benefit, management action names and the value (-1,0 or 1) given to each cell in the table.
4. Visit <http://www.convertcsv.com/csv-to-json.htm> to convert the csv to a JSON string. Upload the csv and select the following options:
   1. Under step 2, **uncheck** the option “First row is column names”
   2. Under step 3, **check** the option “Force Wrap numeric values in double quotes”
   3. In step 5, select “CSV to JSON Array”.
   4. Copy the result, which is a list of lists and must **not** be in quotation marks.
5. With the WHAF app open in firefox, open firebug console, type m= and then paste the list.
6. This list generates two objects that need to be replaced. Each of them is generated using a different function with that list (now referenced as m) as parameter:
   1. **systemBenefits** (associating each system benefit with a component). In firebug console now type: getSystemImpactAttrFromArray(m) and run. Copy the entire, large object without quotation marks and replace the existing **systemBenefits** object in WHAF\_config.js.
   2. **dssParameters** , (which is a very long one,detailing the impact of each management action on every system benefit). In firebug console now type: getParamsFromArray(m) and run. Copy the entire, large object without quotation marks and replace the existing **dssParameters** object, also in WHAF\_config.js.
   3. Both these objects may or may not be beautified for ease of readability.
7. Once the two objects are replaced, the app is ready to go.

For reference, see getDssParams.js: it is not called by the app but includes a sample list of lists; the result of CSV to JSON Array should resemble that list.