# Harvest Errorfile

When harvesting, a list of irregularities is kept.

This is some documentation of the structure in there. It can be used as a supplement to the Reharvest instructions.

* + “count” is just a count of all irregularities
  + “UpToStep” was used to see up to what point errors were tracked. It should be “Finalize”
  + Longdelays checks for differences in timestamps between server time of response and file written time. Should be empty
  + MultiIDs checks whether some ID is used more than once. This could happen if records are updated while the ID-harvest is in progress. The newest one is then used. If there are any, check if the newest ID is really new (that is, the time given in LastUpdate should be after the start of running the script)
  + Disappeared lists records that have disappeared from the results, without giving a ‘deleted record’ back. This is not conforming to the standard, but it happens. If there are many (>100), contact Chris
  + RecSumm gives a list of faults in the downloaded xml-files. This is done by matching a regular expression, as given in Params$RegExErrors.
    - Class tells you which of the regexs did match.
    - Bad tells you what the bad string was
    - Repl means the bad string was replaced with this value
    - Fileno tells you which file. To find the files, check RecFileList (temp) (*timestamp*)
    - Idx means the position in the file where the bad string was found, as a character count.
    - To give some kind of summary, you can use table(Errors$Recsumm$class). This tells you how many errors were found according to which RegEx. In my experience, only numbers 3 and 5 are generated, which are relatively harmless.
    - To get to know the regexes: search the script for Params$RegexErrors, there are some comments about them in the code.
    - This list of irregularities is the largest of all elements, but is mostly harmless. It is mostly included because it tells you in which way the original files were modified
  + RecParse en SummaryOfRecParse are not used when harvesting dc. If they are not empty, we have a problem
  + skippedFiles is a vesctor of filenumbers that have been skipped in the merging step. If it is not empty, we have a problem
  + RecMerge is another check if files are all right. The descr field mostly tells you what the problem is. Duplicate authors/creators are the most common problem, this can be ignored.
  + ColumnsSkipped checks if there are no fields present that have not been processed by the script. If any is present, check first what is in this field, and if it is safe to ignore.
  + RecMergeFinal is a character vector, mostly self-explanatory