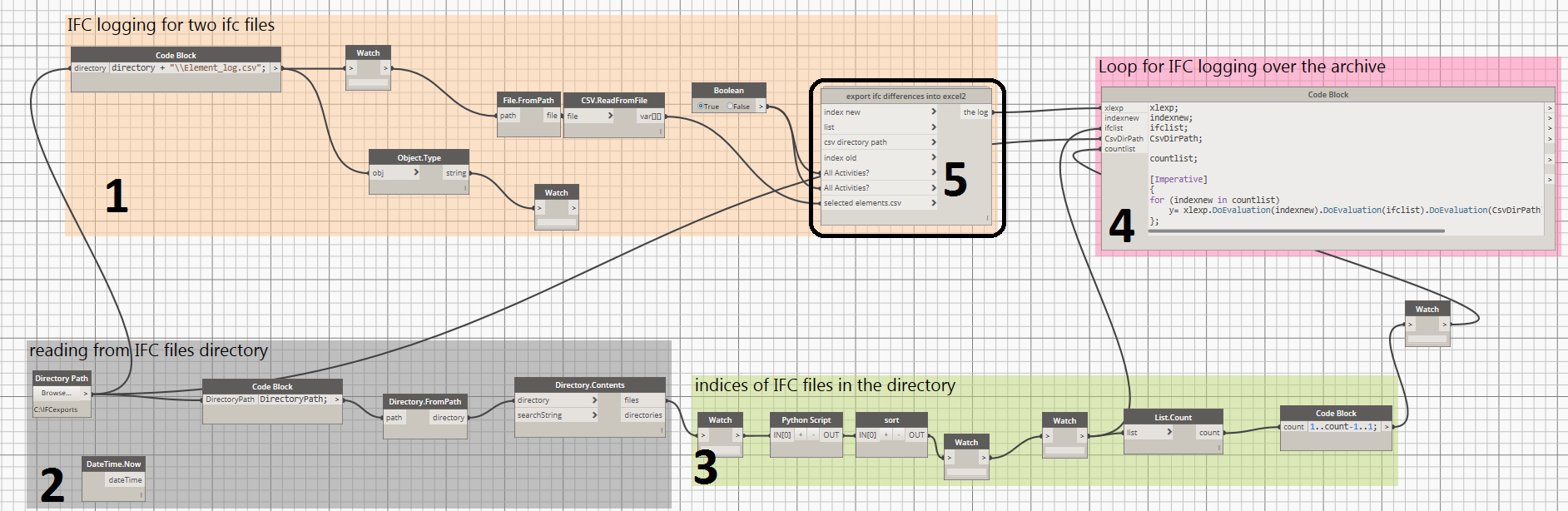
**IFC Logger (Event Log producer)**

**Summary**: A piece of code is written to compare each two consecutive IFC files in our database; the IFC files archive is produced during the design phase using IFC archiver algorithm. Upon every change including adding or removing an IFC element (physical elements such as a wall, window, column, etc. and non-physical elements such as space, zone, etc.) one activity will be saved with the name of “element added” or “element removed” in CSV format. At any point in time and upon BIM manager request, CSV files will be combined using a piece of dynamo code and stored to another CSV file called “the master CSV event log”.

This code can be seen in Figure 1. The relationship between blocks is shown in Figure 2. Blocks are explained as follows:

1. This part is responsible for reading the elements’ IDs and their selection time from the generated CSV file which was generated by block 2 by IFC archiver algorithm. Comparison of two IFC files. The core of this part is the “export IFC differences into excel2” which is numbered as 5.
2. It imports and reads all IFC files from the IFC archive.
3. It creates sets of consecutive IFC files and feeds step 4. Each set contains two contains two consecutive IFC files.
4. It works as a running loop for sets of consecutive IFC files to run block 5 (logging activities into the event log)

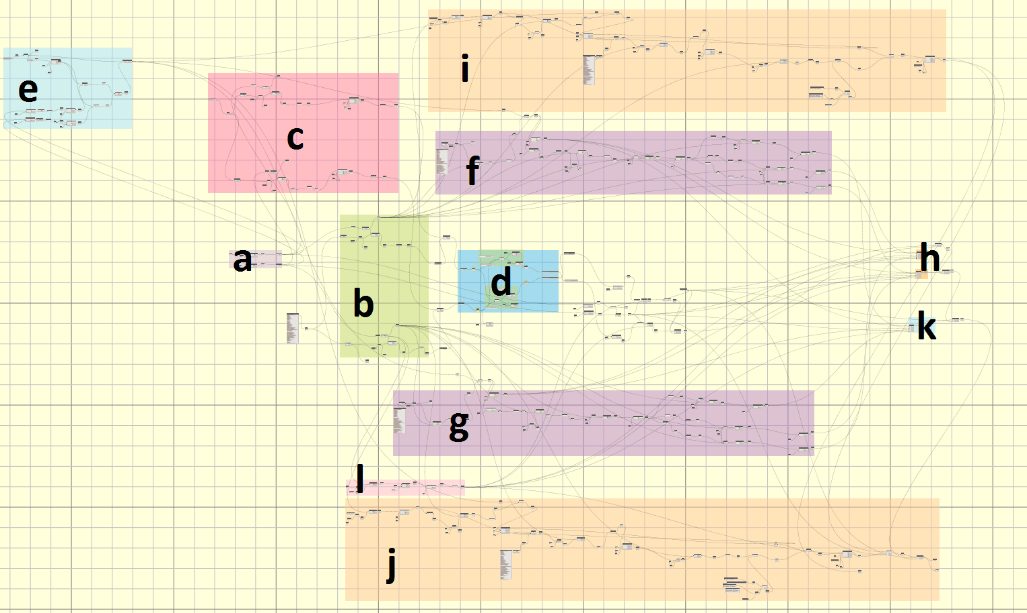


**Figure 1.** IFC Logger algorithm



**Figure 2.** IFC Logger algorithm- the relationship between blocks

1. It is a custom node (a part of step 1) which itself includes different parts shown in Figure 3. In blocks I, f, g and j, the regex search algorithms (for “relocation and rotation” and “property changes”) were used for comparing IFC files. Also, the relationship among blocks is illustrated in Figure 4. In addition, a description for each block is provided as follows:
2. This block is responsible for getting the two consecutive IFC files (“old IFC file” and “new IFC file”) from the IFC archive.
3. This block finds the elements in the IFC files (step a) (which is basically a text file) by using “\u0028\S{23}” regular expression in the GUID based search algorithm.
4. This part is responsible for getting the curtain wall elements by using “\u0028.\*?\n” regular expression in the search algorithm. Since GUID for IFCCURTAINWALL element in IFC tends to vary and is not stable within different IFC files of one BIM model, we used Revit ID (which is fixed) as a trustable signature for IFC comparison.
5. This part is responsible for comparing the list of IDs in two IFC files (output of step b and c) and export the “add and remove” type of activities into the event log.
6. It imports the selected elements and their selection time. It narrows the search query to only the selected elements and, henceforth, optimizes the code. (input for steps i, f, g, and j)
7. text mining pre-work for relocation and rotation detection in the old IFC file (the outputs are the direction and location of the elements in the newer IFC file which considered as inputs for step h and k)
8. text mining pre-work for relocation and rotation detection in the new IFC file (the outputs are the direction and location of the elements in the older IFC file which considered as inputs for step h and k)
9. It compares the direction of the elements in the old (output of step f) and new (output of step g). If they are not equal, it exports a rotation activity into the event log.
10. This combination of nodes gets the property information of the elements in the old IFC file. (the output of this step is one of the inputs of the step j)
11. This combination of nodes gets the property information of the elements in the new IFC file. Then it compares the property information of the element in the old and new IF
12. it compares the location of the element in the old (output of step f) and new (output of step g). If they are not equal, it exports a relocation activity into the event log.
13. It finds the designer information from the new IFC file and set it as “designer” attribute for every type of activity in the event log.



**Figure 3.** IFC files comparer node (node 5) to export the differences between two consecutive IFC files into the CSV format event log



**Figure 4.** IFC comparer node- relationship among blocks of the custom node #5