Duplicate Node Remover

- Duplicate Node Remover Explanation
- Duplicate Node Remover: How to Use

Duplicate Node Remover Explanation

The Duplicate Node Remover is to be used when a customer has had some of their data duplicated, resulting in multiple branches in the tree view that contain mostly the same information. Whether for RPC or production, this tool can be run against their data to resolve that issue. The tool is given a starting point and will recursively crawl through the customer's structure, identifying duplicates at every level and pruning them off of the tree.

Consider a tree with these two folders inside of the same drawer:

```
Folder 1
   |__Folder 2
   |       |__File 1 (10 kb)
   |       |__File 2 (10 kb)
   |       |__File 3 (10 kb)
   |__Folder 3
   |       |__File 4 (10 kb)

Folder 1
   |__Folder 2
   |       |__File 1 (10 kb)
   |       |__File 2 (15 kb)
   |       |__File 5 (10 kb)
   |__Folder 4
   |       |__File 6 (10 kb)
```

If the tool was run against this tree the result would be as follows:

```
Folder 1
   |__Folder 2
   |       |__File 1 (10 kb)
   |       |__File 2 (10 kb)
   |       |__File 2 (1) (15 kb)
   |       |__File 3 (10 kb)
   |       |__File 5 (10 kb)
   |__Folder 3
   |       |__File 4 (10 kb)
   |__Folder 4
   |       |__File 6 (10 kb)
```

What the tool has done is selected the first Folder 1 as the original and then merged the other Folder 1 into it.

- First it removed the duplicate File 1, because it was exactly the same as the original.
- Next, there was a duplicate File 2, but both copies had different sizes, so it renamed the second copy and moved it into the same location as the original.
- It then moved File 5 into the original Folder 2 because it did was a totally unique file.
- Finally, it moved Folder 4 (and all of its contents) into the original Folder 1, because it was a totally unique folder.

This tool will prune the data whether it has been doubled, tripled, or any other number of duplications. It will also traverse any level of folders and sub-folders within the tree.

Duplicate Node Remover: How to Use

1. Unzip the tool and open the folder.
2. Open the appsettings.json file and configure the tool as needed.
    1. DbConnectionConfigurationData is the connection string to access the database where the user's data is kept. By default, this is the connection string that points to localhost:5432 with the default username and password. This should be correct for most local setups.
    2. AccountID is the account which contains the duplicate data that needs to be removed.
    3. FirstRootNode and SecondRootNode are optional values.
        - If both are set to null, then the tool will run from the account's root node and cover the entire structure.
        - If just one of them is set to the ID of a node, then the tool will start at that part of the tree and prune all of its children.
        - If both are set to the ID of a node, then the tool will bring all of the contents of the SecondRootNode into the FirstRootNode, removing duplicates along the way. The second node will still be left in the tree structure, but it will be empty.
    4. WriteToFile will tell the tool to write all of the nodes to be edited to file. There will be two files produced: moveRename.xml and remove.xml. This option is available because the tool may run for a very long time on large datasets, and this allows you to break the process into gathering all of the nodes to update and executing changes on them.
    5. LoadFromFile will circumvent the FirstRootNode and SecondRootNode and instead load all of the files from the moveRename.xml and remove.xml files.
    6. DoPruningStep determines whether the tool should attempt to make all of the changes or not. If you want to break up the process of cleaning a user's data you could do it in two passes with the following configurations:
        1. WriteToFile is true, LoadFromFile is false, DoPruningStep is false. This will do the work of crawling through the user's data, extract the nodes to be updated, write them to file, and then terminate.
        2. WriteToFile is false, LoadFromFile is true, DoPruningStep is true. This will load the needed changes from file without having to crawl through the data again, then execute those changes.
3. **Back up the user's data before running the tool.**
4. Run DuplicateNodeRemover.exe. It will proceed in two main steps:
    1. Finding all of the nodes to update. This might take considerable time depending on the size of the data to be pruned. Eventually it will report how many nodes are set to be removed, moved, and renamed-then-moved.

    2. Executing all of the changes. If there are more than a thousand nodes to update, you will have the option to periodically decide if the tool should continue or if it should save its progress and resume later.
5. Done!