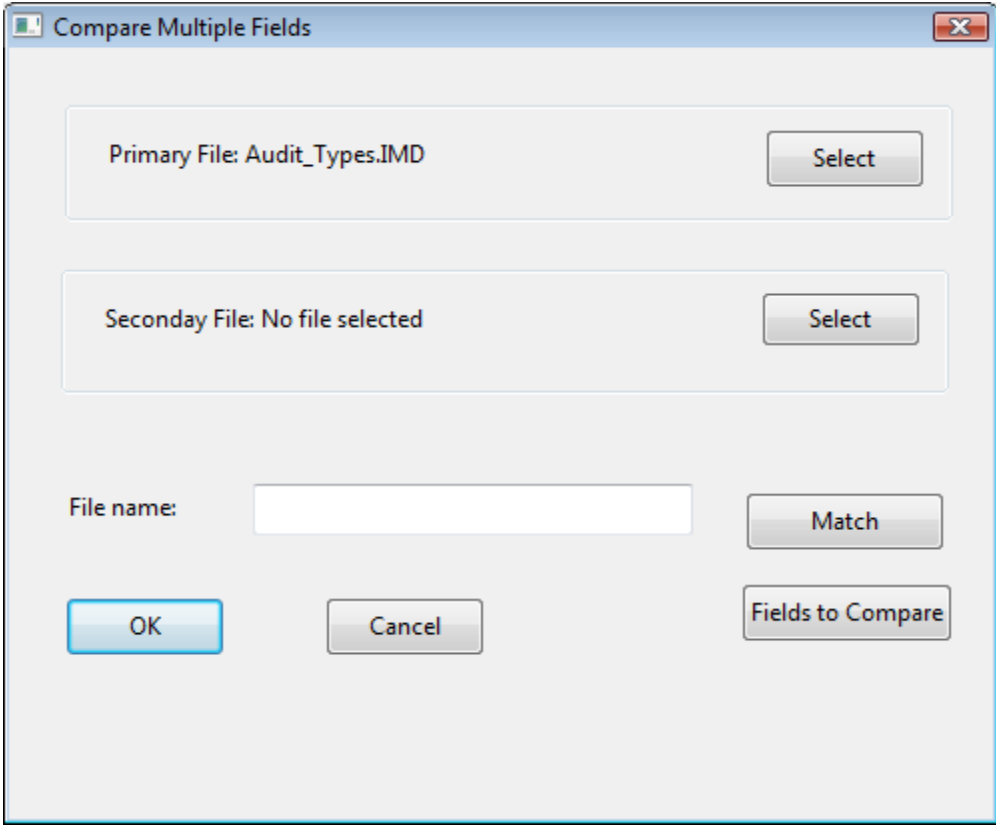


Purpose of script: This script allows the user to compare the same file at two points in time or two files that are supposed to be the same. The script will first join the two files together using matching fields using the "Matches only" join function and then creates a virtual field in the joined file to compare the fields that the user has selected. The comparison can be performed on character, numeric and date fields.

When starting the script the user will see the following menu. The menu allows you to supply the primary file and secondary file (much in the same way as selected the primary and secondary file using a join). You can not join a file with itself.

The fields names in both files must both be the same and the types must also be the same. If the names or types are different the script will give an error.

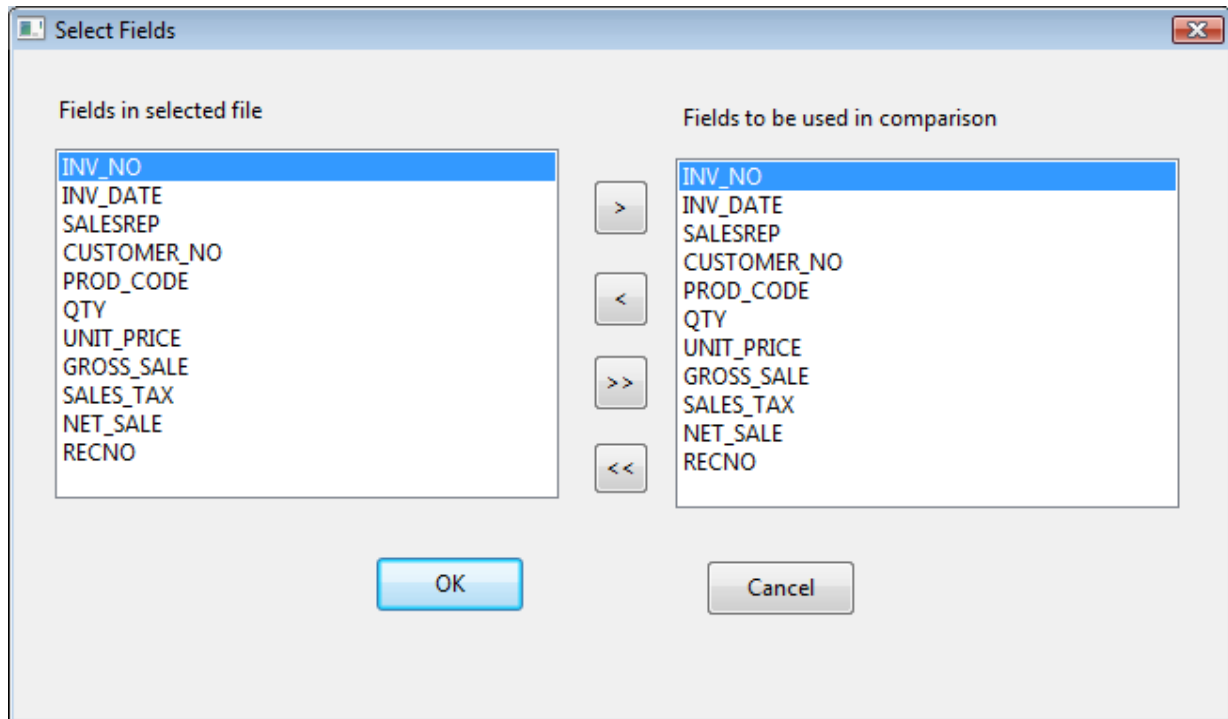


The next option is to select the fields to match on by clicking on the Match button. See next page for the menu that opens up.

The image shows a dialog box titled "Select Matches" with a close button in the top right corner. Inside the dialog, there are eight rows, each labeled "1st Match" through "8th Match". Each row contains a horizontal dropdown menu. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

You may choose up to 8 fields to match on. Each field must be present in both files with the same type. All fields can be left blank except for the 1st match which must be selected. There must be at least one common field. If there are no common fields and the files are the same you can create a virtual field in both files using the @recno() function and join base on that field. This will only work if the files have the same number of transactions and are in the same order.

The final menu allows you to select the fields to use in the comparison.



The comparison can be done on character, numeric or date fields. The field name and type must be the same in each file. The default is to have all fields selected for comparison.

The final step is to choose a file name for the completed file.

The script will then join the two files together based on the join fields. It will then create a new field for each compare, the new field will be called COMPARE_ old field name.

The output would be the following:

- For a numeric field it would subtract one field from the other, so any changes would be reflected by having a non 0 as a result.
- For character fields it uses the @CompareNoCase() function, so any changes would be reflected by having a non 0 as a result.
- For date fields it uses @Age() function, so any changes would be reflected by having a non 0 as a result.

In order to find out which fields have changed you need to filter out all non 0 amounts in the new COMPARE_ fields.

A history is added that show the name of the primary and secondary file, the fields used to match and the fields used for comparison.

This script was developed by the GEL Group that is made up of Sunder Gee, Brian Element and Steve Luciani. If you have any comments or problems please drop me a note at elementb@rogers.com.