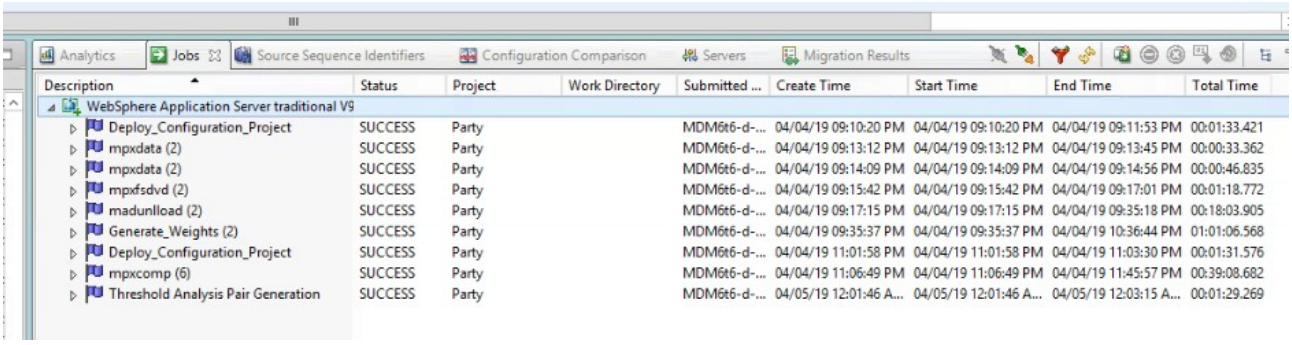
Table of Contents

PME Process For PME Tuning : Specifically Targetted for MDM AE Projects

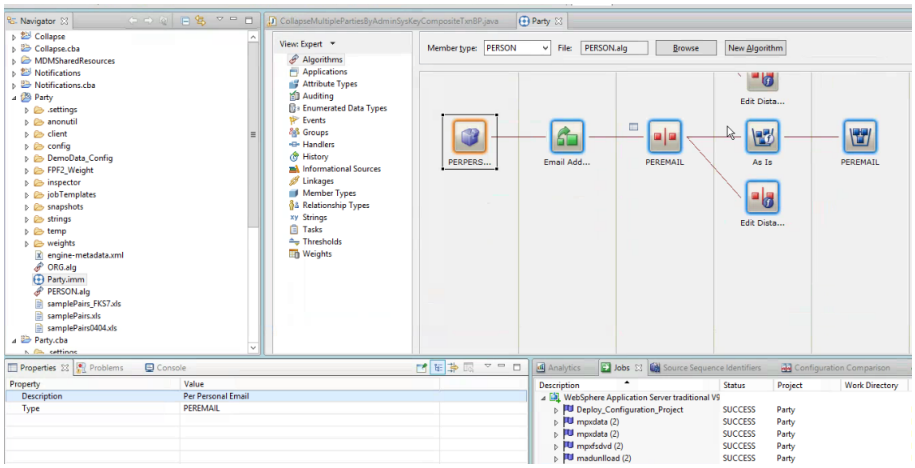
Ian Dallas
iandallas@uk.ibm.com

1 Summary of all PME jobs :

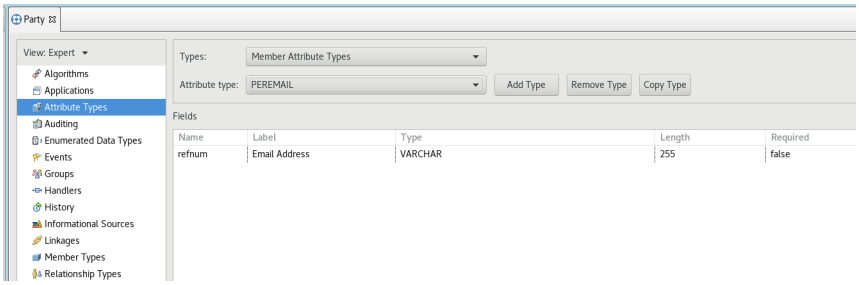


2 Edit algorithm : eg Add Personal Email

Create a new swim lane containing person email PEREMAIL (and optionally other email types as required). If used for Householding also add a comparison step.

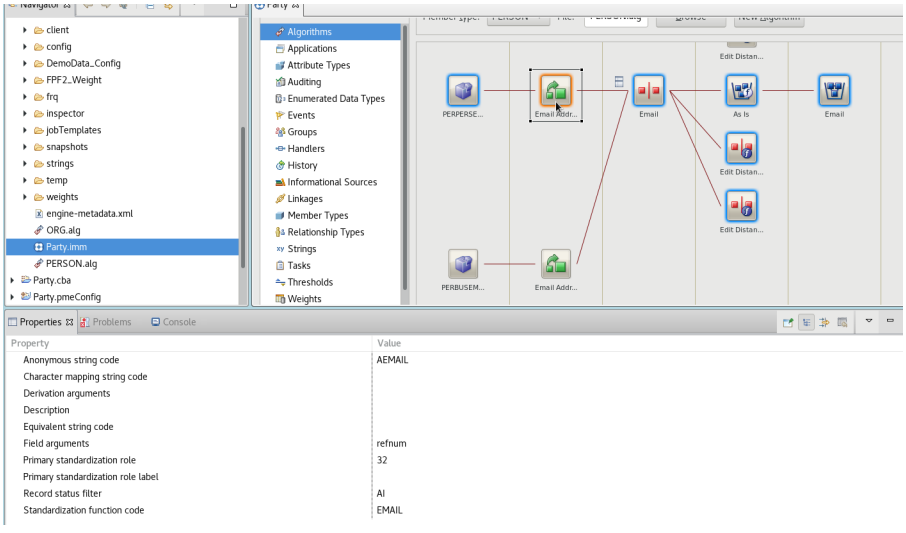


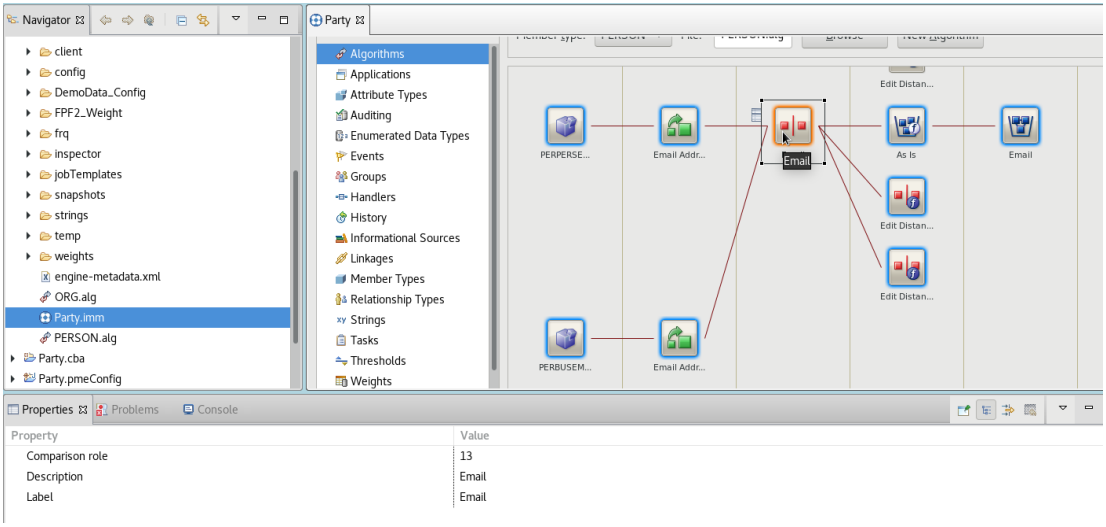
Change the PEREMAIL attribute type to “refnum” : this is required by MDM AE because email address is supplied using PartyContactMethodBObj “refnumber” field (Usage Type 7)

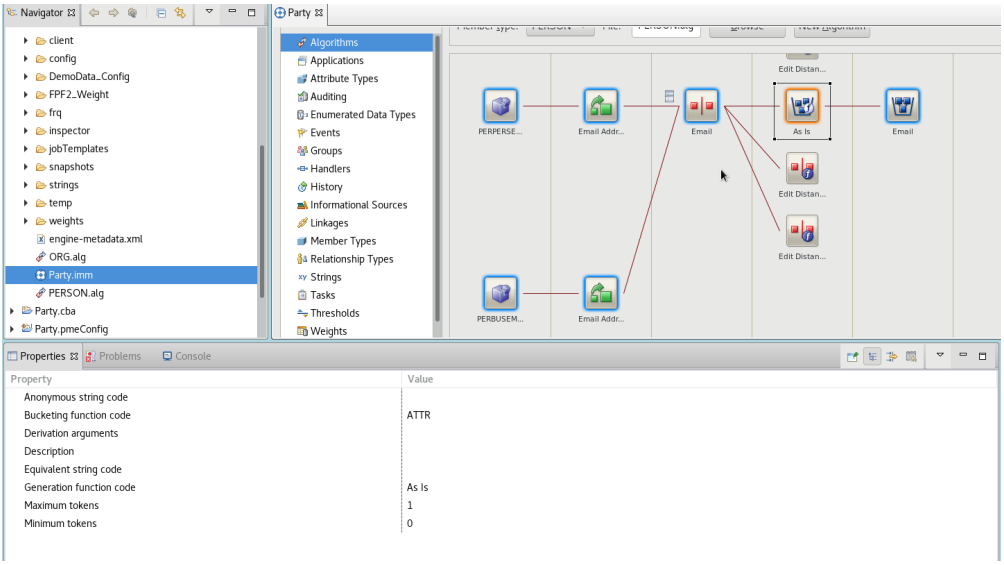


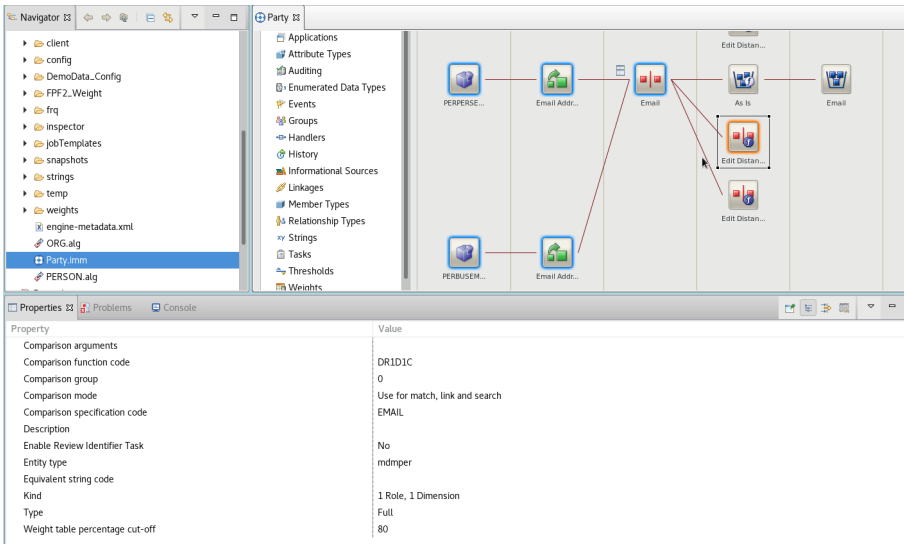
Now set the properties of the various elements in the new swim lane.

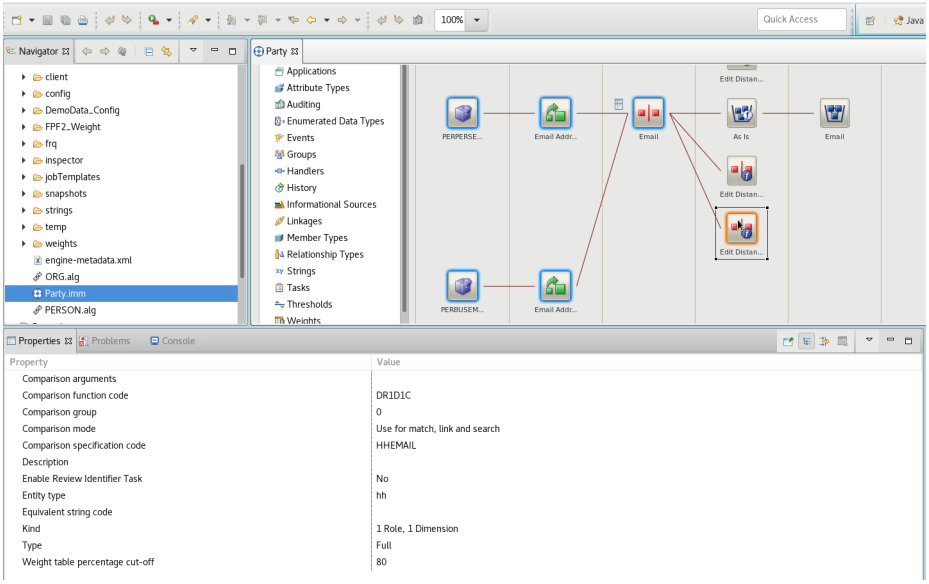
Set field arguments to “refnum” for the email Standardiser

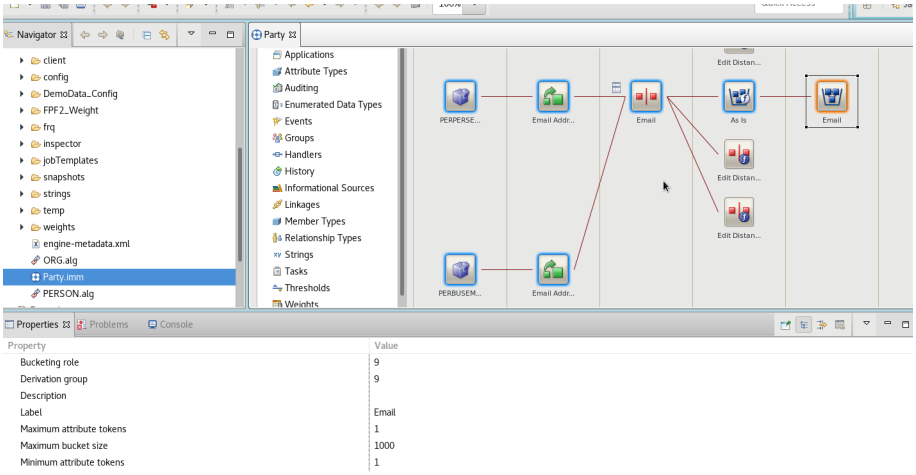



Navigator
client
config
DemoData_Config
FPF2_Weight
frq
inspector
jobTemplates
snapshots
strings
temp
weights
engine-metadata.xml
ORG.alg
Party.imm
PERSON.alg
Party.cba
Party.pmeConfig
Party
Algorithms
Applications
Attribute Types
Auditing
Enumerated Data Types
Events
Groups
Handlers
History
Informational Sources
Linkages
Member Types
Relationship Types
Strings
Tasks
Thresholds
Weights
PERPERSE...
Email Addr...
Email
As Is
Edit Distan...
PERBUSEM...
Properties
Problems
Console
Property
Value
Comparison role
13
Description
Email
Label
Email

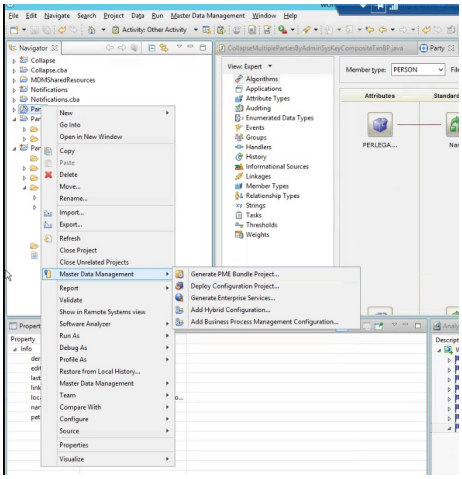

Navigator
Party
Algorithms
Applications
Attribute Types
Auditing
Enumerated Data Types
Events
Groups
Handlers
History
Informational Sources
Linkages
Member Types
Relationship Types
Strings
Tasks
Thresholds
Weights
Edit Distan...
PERPERSE...
Email Addr...
Email
As Is
PERBUSEM...
Properties
Problems
Console
Property
Value
Anonymous string code
Bucketing function code
ATTR
Derivation arguments
Description
Equivalent string code
Generation function code
As Is
Maximum tokens
1
Minimum tokens
0


Navigator
Party
Applications
Attribute Types
Auditing
Enumerated Data Types
Events
Groups
Handlers
History
Informational Sources
Linkages
Member Types
Relationship Types
Strings
Tasks
Thresholds
Weights
Edit Distan...
PERPERSE...
Email Addr...
Email
As Is
PERBUSEM...
Properties
Problems
Console
Property
Value
Comparison arguments
Comparison function code
DR1D1C
Comparison group
0
Comparison mode
Use for match, link and search
Comparison specification code
EMAIL
Description
Enable Review Identifier Task
No
Entity type
mdmper
Equivalent string code
Kind
1 Role, 1 Dimension
Type
Full
Weight table percentage cut-off
80


Navigator
client
config
DemoData_Config
FPF2_Weight
frq
inspector
jobTemplates
snapshots
strings
temp
weights
engine-metadata.xml
ORG.alg
Party.imm
PERSON.alg
Party
Applications
Attribute Types
Auditing
Enumerated Data Types
Events
Groups
Handlers
History
Informational Sources
Linkages
Member Types
Relationship Types
Strings
Tasks
Thresholds
Weights
PERPERSE...
Email Addr...
Email
As Is
Email
Edit Distan...
PERBUSEM...
Properties
Problems
Console
Property
Value
Comparison arguments
Comparison function code
DR1D1C
Comparison group
0
Comparison mode
Use for match, link and search
Comparison specification code
HHEMAIL
Description
Enable Review Identifier Task
No
Entity type
hh
Equivalent string code
Kind
1 Role, 1 Dimension
Type
Full
Weight table percentage cut-off
80

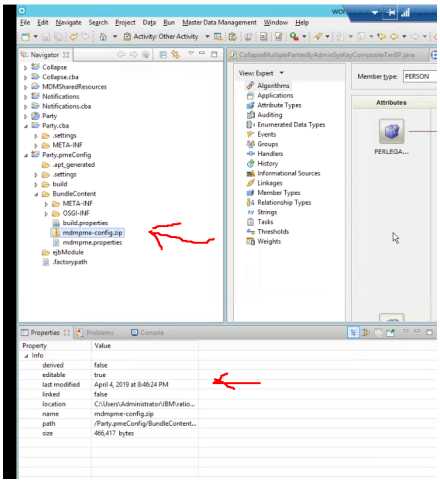

Navigator
client
config
DemoData_Config
FPF2_Weight
frq
inspector
jobTemplates
snapshots
strings
temp
weights
engine-metadata.xml
ORG.alg
Party.imm
PERSON.alg
Party
Applications
Attribute Types
Auditing
Enumerated Data Types
Events
Groups
Handlers
History
Informational Sources
Linkages
Member Types
Relationship Types
Strings
Tasks
Thresholds
Weights
PERPERSE...
Email Addr...
Email
As Is
Email
Edit Distan...
PERBUSEM...
Properties
Problems
Console
Property
Value
Bucketing role
9
Derivation group
9
Description
Label
Email
Maximum attribute tokens
1
Maximum bucket size
1000
Minimum attribute tokens
1

Note : if you have already created the and associated Party.cba projects before altering the algorithm, then you will need to delete them and re-create.



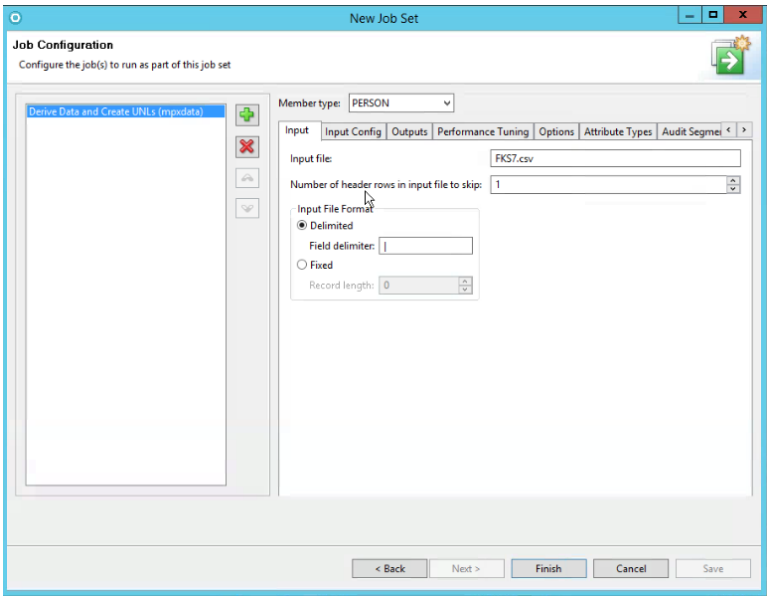
This is because the pmeConfig project with the mdmpme-config.zip file is only created once

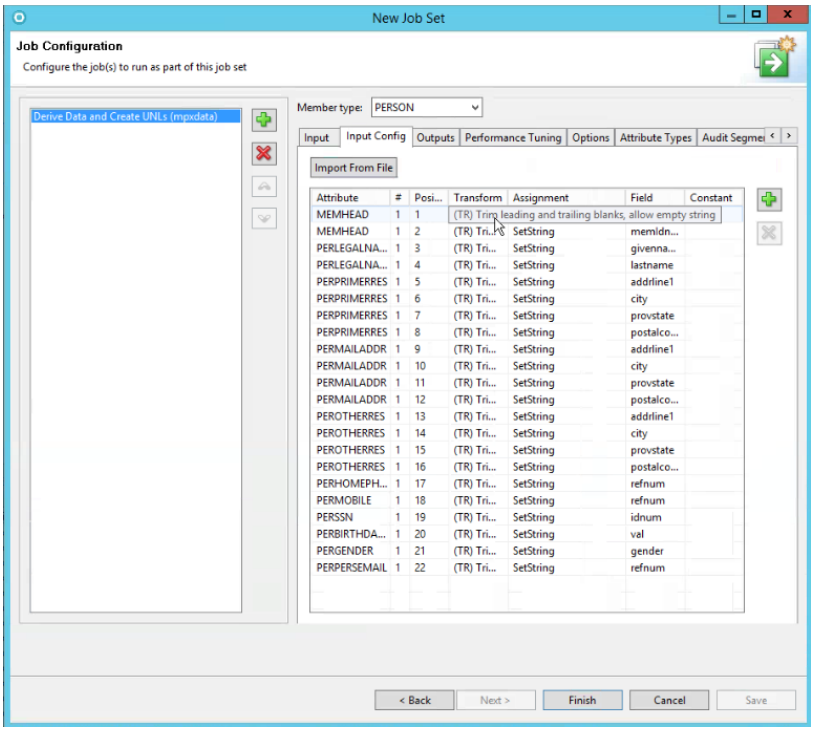
AND IT IS NEVER UPDATED WHEN YOU CHANGE THE ALGORITHM.

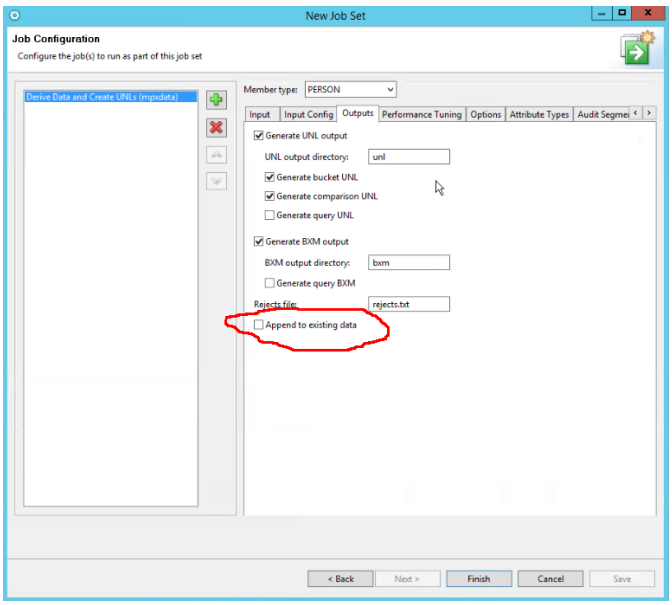


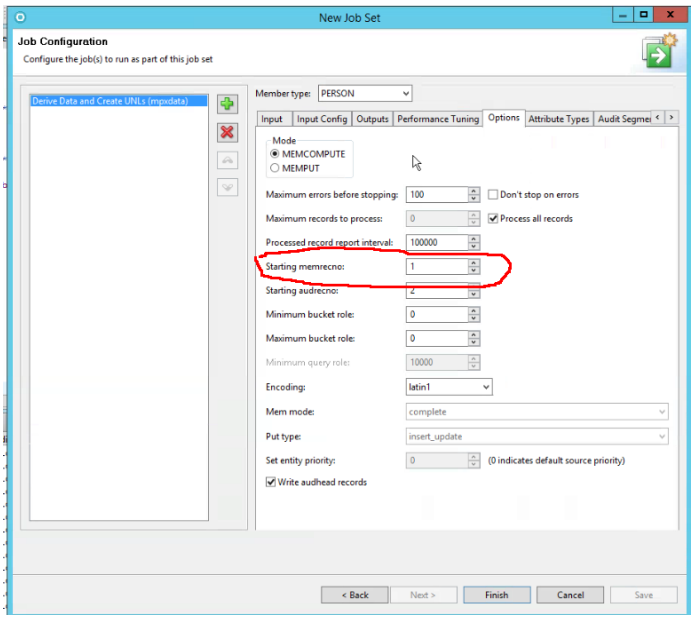
3 data load #1: mpxdata

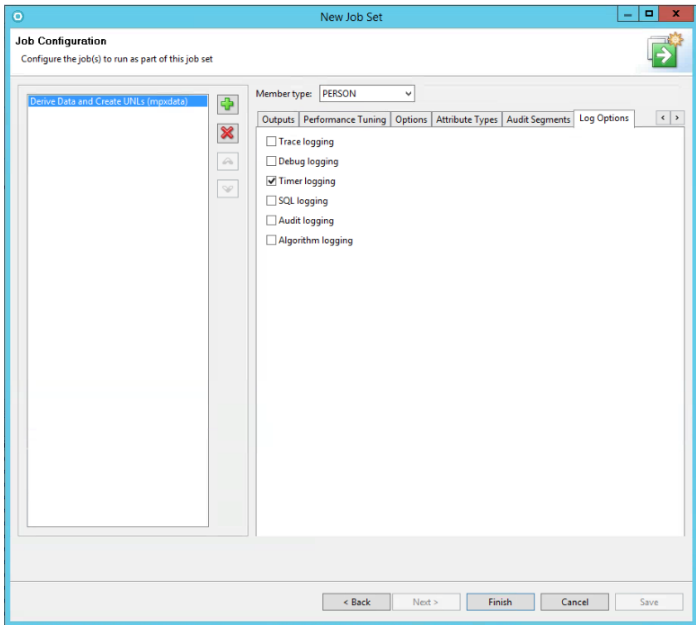
Load data from the 1st data source.





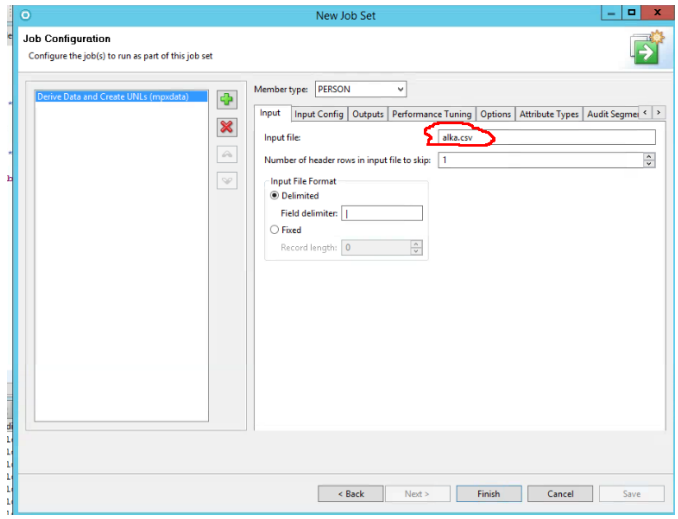

New Job Set
Job Configuration
Configure the job(s) to run as part of this job set
Derive Data and Create UNLs (mpxdata)
Member type: PERSON
Input
Input Config
Outputs
Performance Tuning
Options
Attribute Types
Audit Segmei
Generate UNL output
UNL output directory: unl
Generate bucket UNL
Generate comparison UNL
Generate query UNL
Generate BXM output
BXM output directory: bxm
Generate query BXM
Rejects file: rejects.txt
Append to existing data
< Back
Next >
Finish
Cancel
Save


New Job Set
Job Configuration
Configure the job(s) to run as part of this job set
Derive Data and Create UNLs (mpxdata)
Member type: PERSON
Input
Input Config
Outputs
Performance Tuning
Options
Attribute Types
Audit Segmei
Mode
MEMCOMPUTE
MEMPUT
Maximum errors before stopping: 100
Don't stop on errors
Maximum records to process: 0
Process all records
Processed record report interval: 100000
Starting memrecno: 1
Starting audrecno: 2
Minimum bucket role: 0
Maximum bucket role: 0
Minimum query role: 10000
Encoding: latin1
Mem mode: complete
Put type: insert_update
Set entity priority: 0
(0 indicates default source priority)
Write audhead records
< Back
Next >
Finish
Cancel
Save

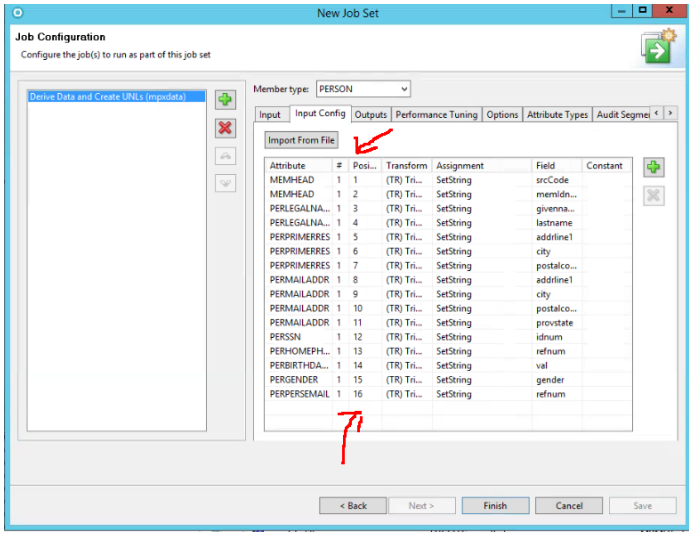

New Job Set
Job Configuration
Configure the job(s) to run as part of this job set
Derive Data and Create UNLs (mpxdata)
Member type: PERSON
Outputs
Performance Tuning
Options
Attribute Types
Audit Segments
Log Options
Trace logging
Debug logging
Timer logging
SQL logging
Audit logging
Algorithm logging
< Back
Next >
Finish
Cancel
Save

4 data load #2 mpxdata

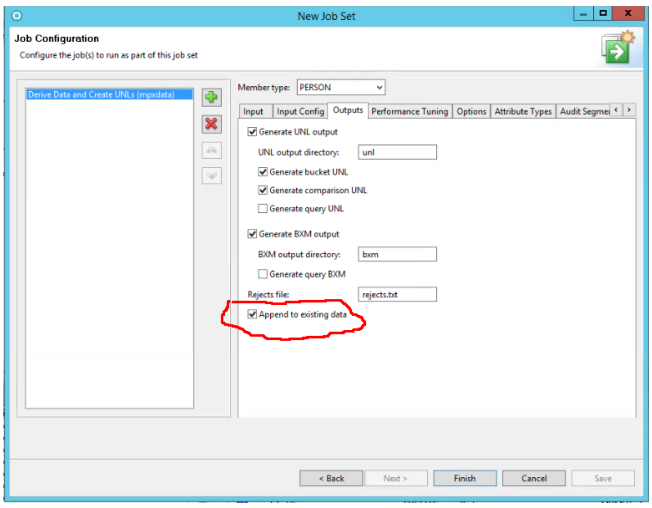
Load data from the 2nd data source.



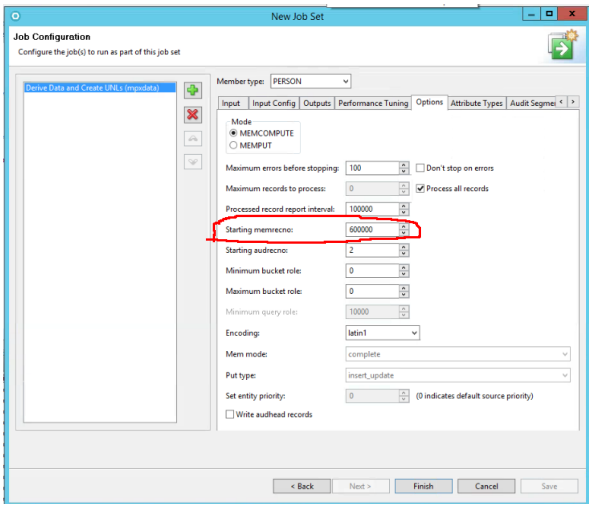
Be careful about the Position numbers : they can get messed up!

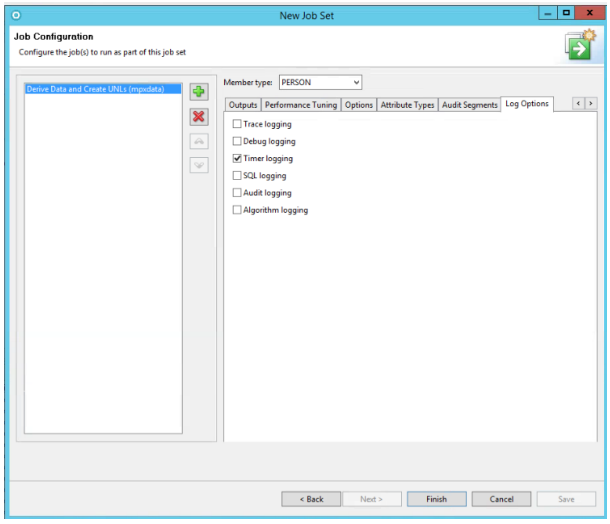


Append to existing data so UNL files will have both data from both 1st and 2nd data sources



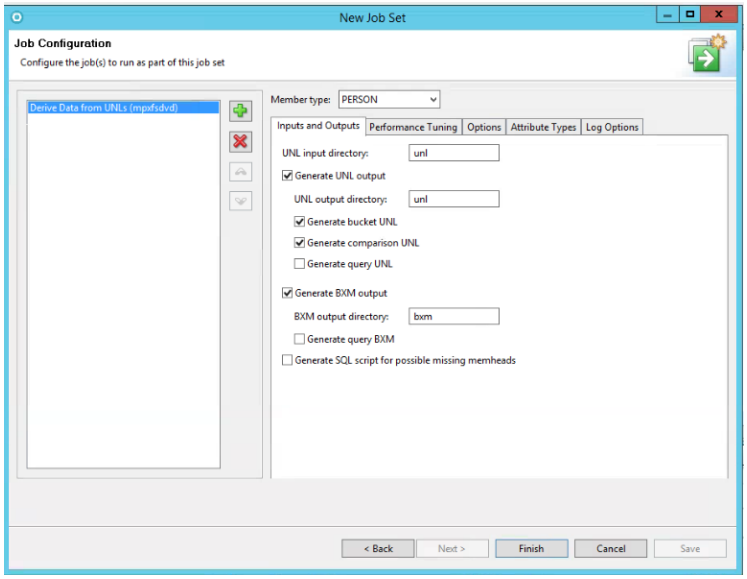
Be careful to set the “Starting memrecnum” so that is is slightly higher figure than the number of records read from the 1st data source. This will avoid error with duplicate rec numbers being loaded to the database in the next steps..





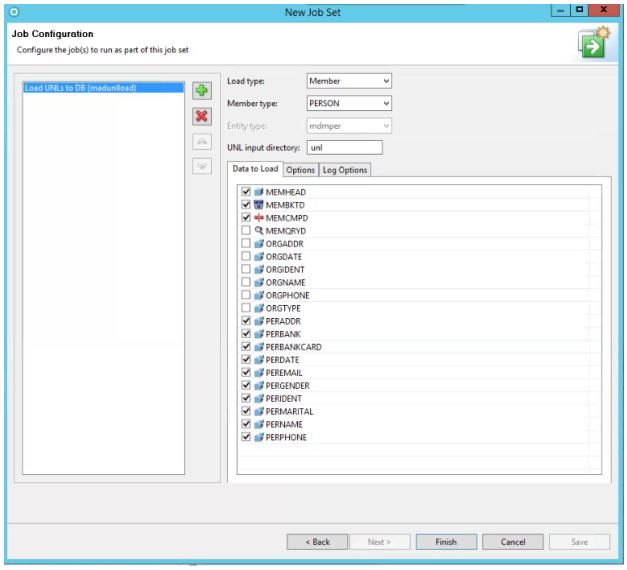
5 Generate BXM Data mpxfsdvd

Because each mxpdata load overwrites the BXM data, we need to re-run to populate the BXM load files with data from both data sources.

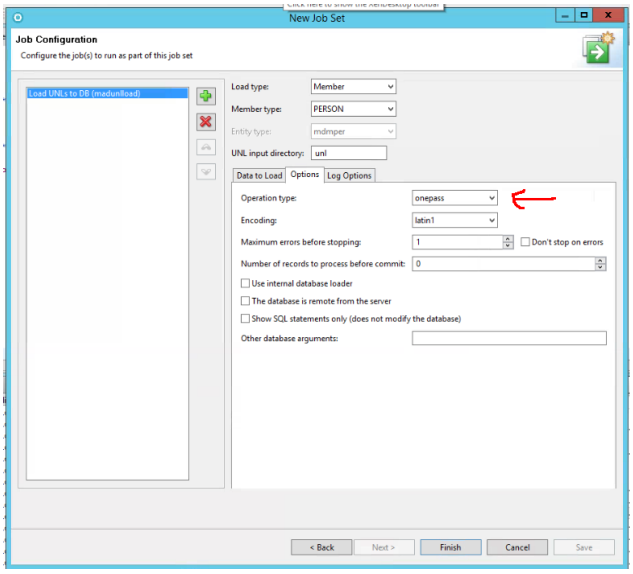


6 load data files to MDM DB : madunlload

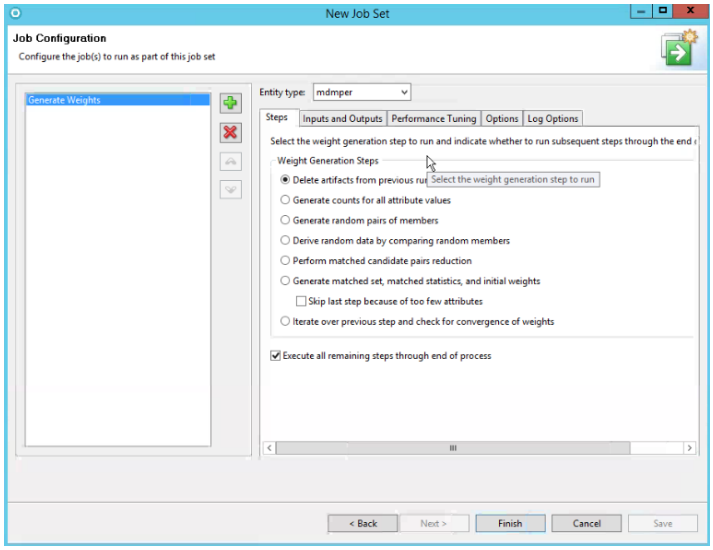
Load the data, buckets etc into the MDM database.



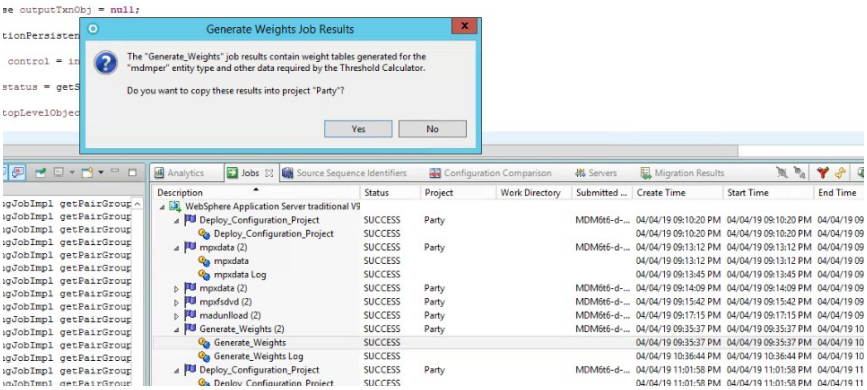
keep default “onepass” option : re-creates the virtual database tables, so don’t worry if previously loaded data is there.. it will be replaced.



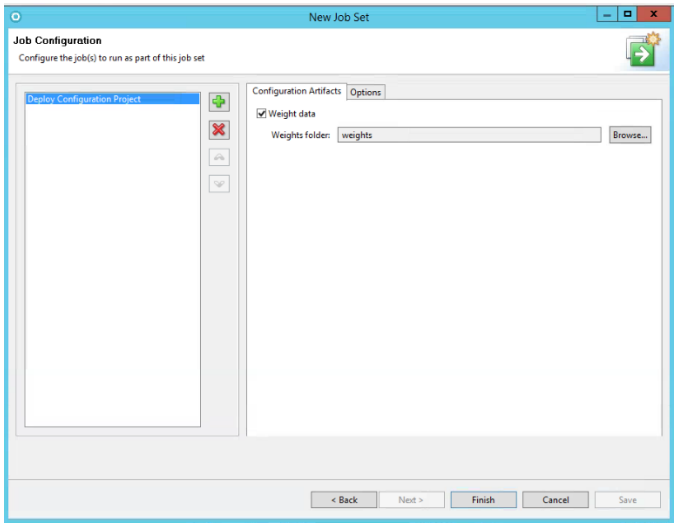
7 Generate Weights



let the process finish, then right clock, “get job results” and save the weights in to the Party workspace project in the weights folder



Deploy the configuration again ie new weights get deployed:

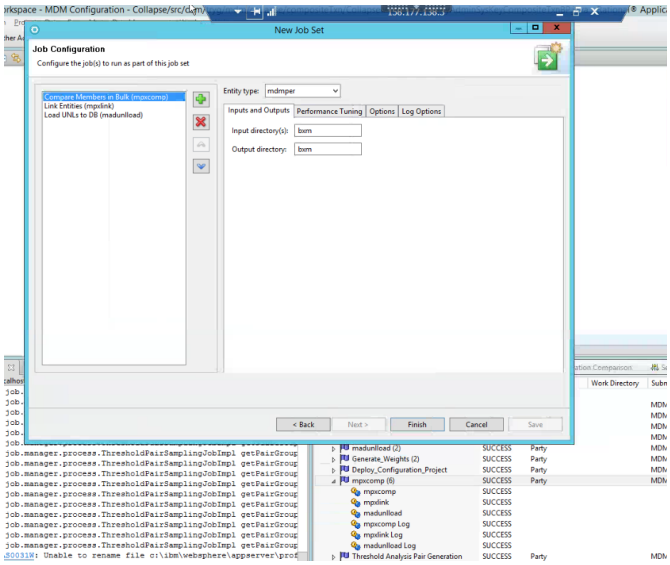


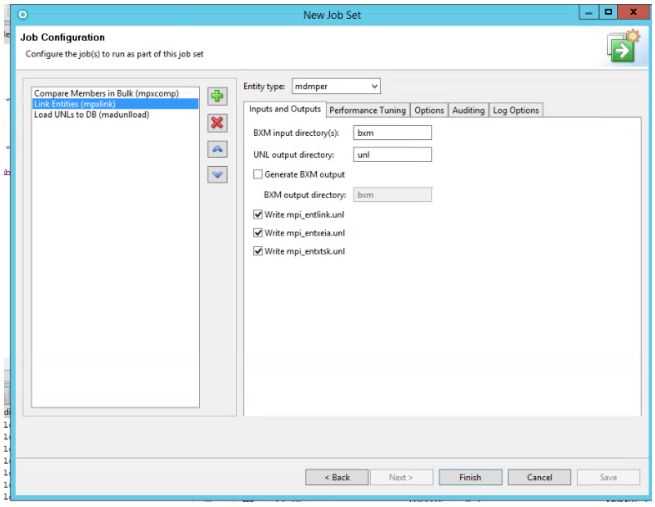
Some of the algorithm “swim lanes”, like business email, if not populated, will error out the weight gen process. So either remove un-needed swim lanes, or change its weight threshold to “1”.

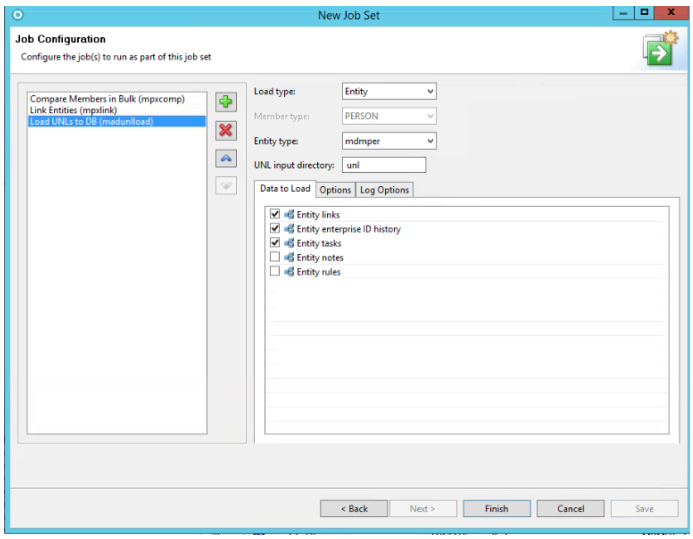
Note : every change to the algorithm, must be followed by a Deploy Configuration and madunlload. Re-run Weight Generation as required.

8 Run the Bulk Cross Match processes

(3 of them : mpxcomp mpxlink madunload): combine all 3 jobs and save as a job template (just like recommended for all the other single job processes)







9 Perform Analysis Pairs Generation..

When this process is completed, get the job results, and save as an XLS file which can most easily be analysed/processed using Excel.