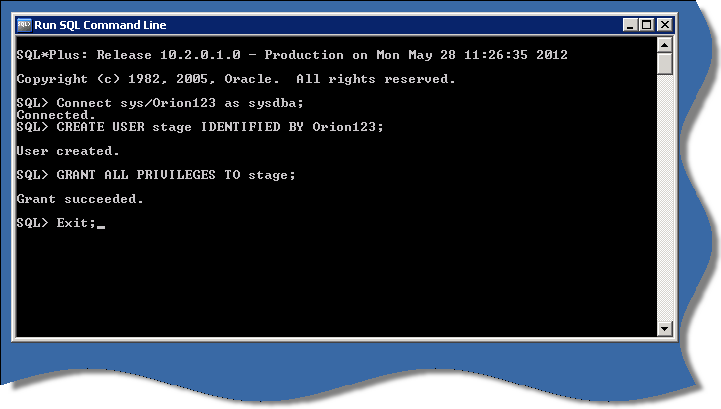
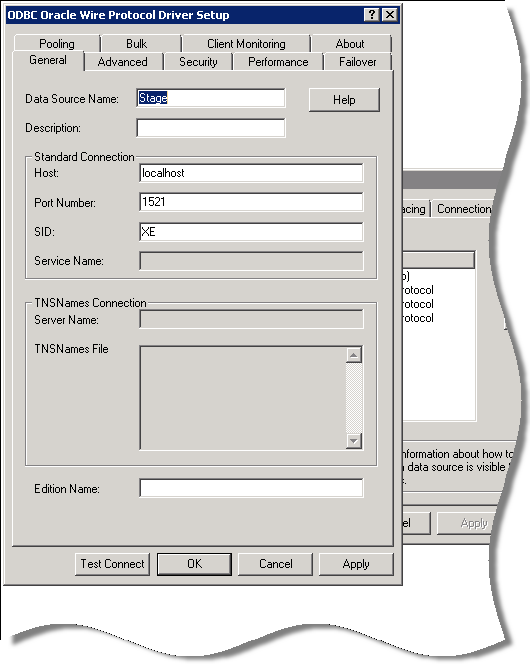
**Create New Entity in qMDM**

Assume that qMDM 3.2 is installed and running on the RACE qMDM training image.

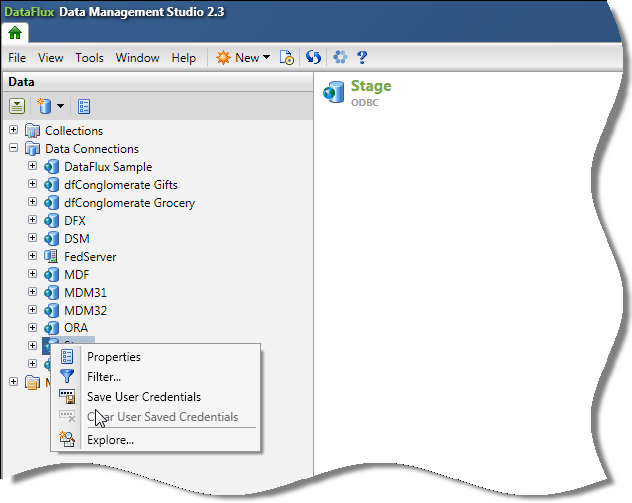
1. **Setup Staging Database to prepare loading the data**
   1. Create staging database called *Stage*
      1. Start SQLPlus and run the following commands:
         1. Connect sys/Orion123 as sysdba;
         2. CREATE USER *stage* IDENTIFIED BY *Orion123*;
         3. GRANT ALL PRIVILEGES TO stage;
         4. Exit;



* 1. Create ODBC DSN for staging database:
     1. DSN: Stage
     2. Host: localhost
     3. Port Number: 1521
     4. SID: xe



* 1. Save user credentials in DM Studio



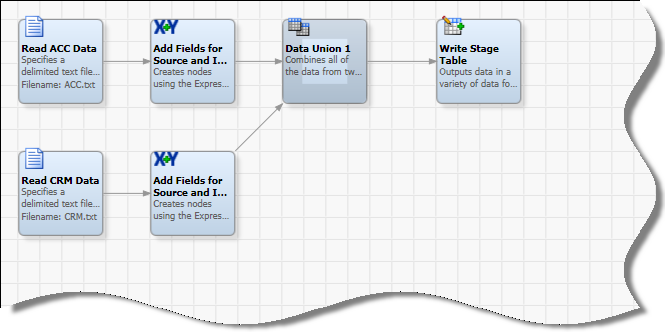
1. **Preparing Source Data -** Load data into staging database
   1. There are two data source files. Create a data job to load both files into one table using these column names:

|  |  |  |
| --- | --- | --- |
| **Source: CRM** | **Source: ACC** | **Stage Table: POC\_DATA** |
| SOURCE | SOURCE | **SOURCE** |
| ID | PERSON\_ID | REC\_ID |
| ISO\_LOCALE | ISO\_LOCALE | **ISO\_LOCALE** |
| TITLE | TITLE | TITLE |
| FIRST\_GIVEN\_NAME | FIRST\_NAME | FIRST\_NAME |
| FAMILY\_NAME | SURENAME | LAST\_NAME |
| ADDRESS\_LINE\_1 | ADDRESS\_LINE\_1 | ADR\_LINE\_1 |
| ADDRESS\_LINE\_2 |  | ADR\_LINE\_2 |
| TOWN | CITY | TOWN |
| COUNTY |  | COUNTY |
| POSTCODE | ZIP | POSTCODE |
| COUNTRY | COUNTRY | COUNTRY |
| EMAIL\_ADDRESS | EMAIL | EMAIL |
| HOME\_PHONE\_NUMBER | PHONE\_HOME | HOME\_PHONE |
| MOBILE\_PHONE\_NUMBER | PHONE\_MOBILE | MOBILE\_PHONE |
| WORK\_PHONE\_NUMBER | PHONE\_WORK | WORK\_PHONE |
| DATE\_OF\_BIRTH | BIRTHDAY | DOB |
| GENDER | GENDER | GENDER |
| COMPANY\_ID | COMPANY\_ID | COMPANY\_ID |

* 1. Add two extra fields to the source data and set the values to indicate the Source and the Country:

**SOURCE** => **ACC\_UK** or **CRM\_UK** respectively

**ISO\_LOCALE** => **GBR** to indicate that it is UK data



*Job to load POC Data*

**Note:** We could also load the data into separate tables and build a SQL Join to load the data

* 1. Run the job to load the data into the staging database.
  2. *Profile the data to get a better understanding of the data*
  3. *Identify the attribute which are going to build the Entity*

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Attribute** | **Type** | **Length** |
| Person | TITLE | String | 15 |
| Person | FIRST\_NAME | String | 50 |
| Person | LAST\_NAME | String | 50 |
| Person | DOB | Date |  |
| Person | GENDER | String | 2 |
| Person | EMAIL | String | 100 |
| Person | HOME\_PHONE | String | 20 |
| Person | MOBILE\_PHONE | String | 20 |
| Person | WORK\_PHONE | String | 20 |
| Person | COMPANY\_ID | String | 5 |
| Address | ADR\_LINE\_1 | String | 100 |
| Address | ADR\_LINE\_2 | String | 100 |
| Address | TOWN | String | 50 |
| Address | COUNTY | String | 30 |
| Address | POSTCODE | String | 10 |
| Address | COUNTRY | String | 30 |

**We will split it into to two Entity Types. Address is going to be an abstract Entity Type and Person a non abstract Entity Type which has Address as its Parent Entity Type.**

* 1. Steps 2.3 and 2.4 can help us to identify the attributes that we are going to use to set Match Conditions to identify duplicates and fields we want to standardize and cleanse.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | C1 | C2 | C4 | C5 | C6 |
| DF\_FIRST\_NAME\_MC | X |  | X | X | X |
| DF\_LAST\_NAME\_MC |  | X | X | X | X |
| DF\_ADR\_LINE\_1\_MC | X | X |  |  |  |
| DF\_POSTCODE\_MC | X | X | X | X | X |
| DOB | X | X |  |  | X |
| DF\_MOBILE\_PHONE\_STND |  |  | X |  |  |
| DF\_EMAIL\_STND |  |  |  | X |  |
| DF\_WORK\_PHONE\_STND |  |  |  |  |  |
| DF\_HOME\_PHONE\_STND |  |  |  |  |  |
| DF\_TITLE\_STND |  |  |  |  |  |
| DF\_POSTCODE\_STND |  |  |  |  |  |

**Tip:** Use an Excel [spreadsheet](file:///D:\qMDM_Workshop\Docs\Entity_Design.xlsx) to describe the entity with its attributes, cleansing and standardization rules and match conditions.

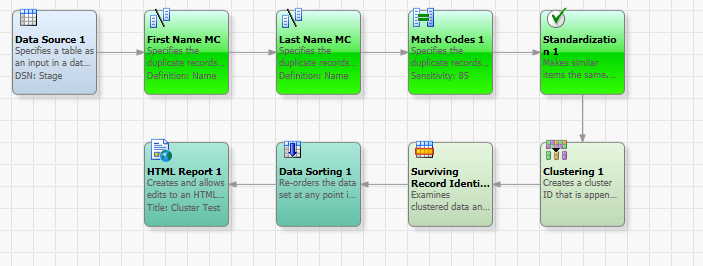
* 1. **Test Match Condition and standardization**

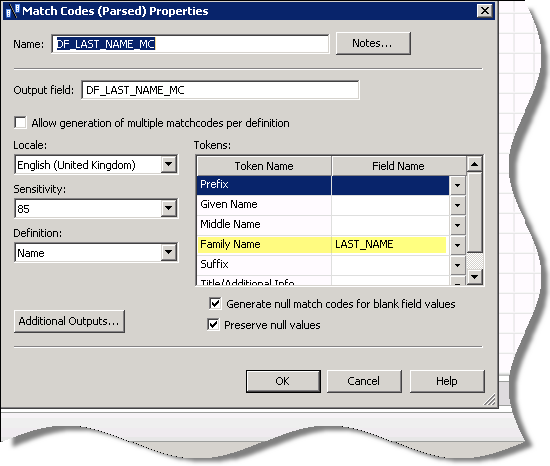
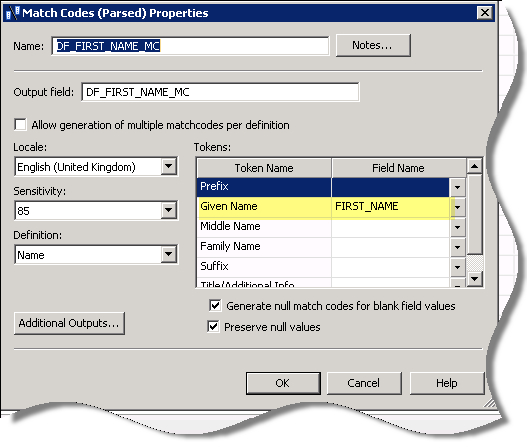
At this stage best practice is to check if the rules for standardization, cleansing as well as the match conditions are right to find out what match code sensitivity to use. In order to do so we will create a test job in DM Studio this will help us later to complete the stnd job for the entity:

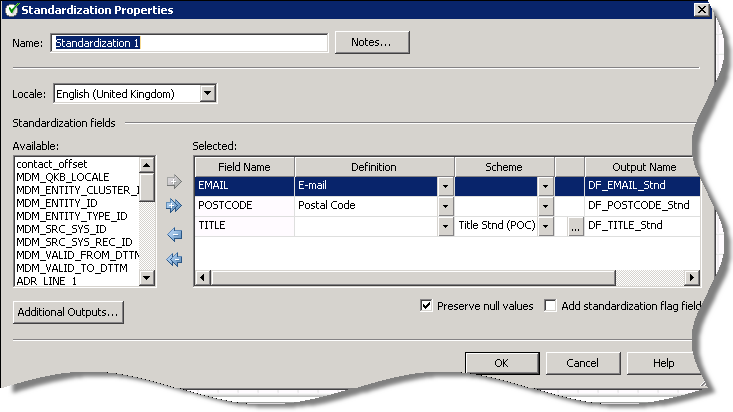
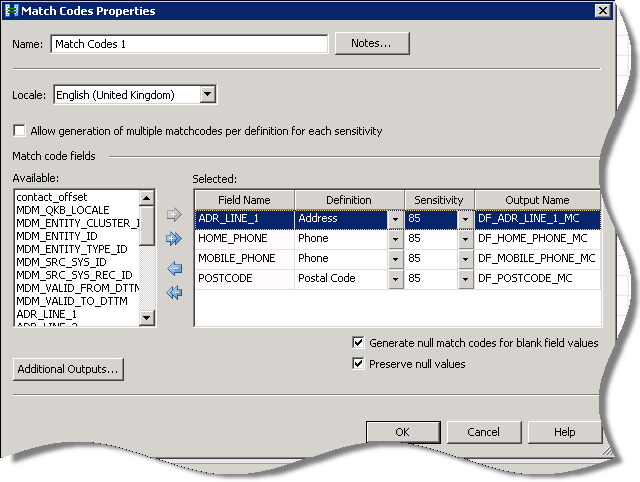
* + 1. Create a new job in DM Studio: stnd\_test
    2. Read staging data
    3. Create Match Codes and Standardization Fields (using the Entity field names already!)

|  |  |  |  |
| --- | --- | --- | --- |
| **Flux Attribute** | **Definition** | **Selectivity** | **Scheme** |
| DF\_FIRST\_NAME\_MC | Name | 85 |  |
| DF\_LAST\_NAME\_MC | Name | 85 |  |
| DF\_ADR\_LINE\_1\_MC | Address | 85 |  |
| DF\_POSTCODE\_MC | Postcode | 85 |  |
| DF\_HOME\_PHONE\_STND | Phone |  |  |
| DF\_MOBILE\_PHONE\_STND | Phone |  |  |
| DF\_WORK\_PHONE\_STND | Phone |  |  |
| DF\_EMAIL\_STND | e-mail |  |  |
| DF\_POSTCODE\_STND | Postcode |  |  |
| DF\_TITLE\_STND |  |  | Title Stnd (POC) |

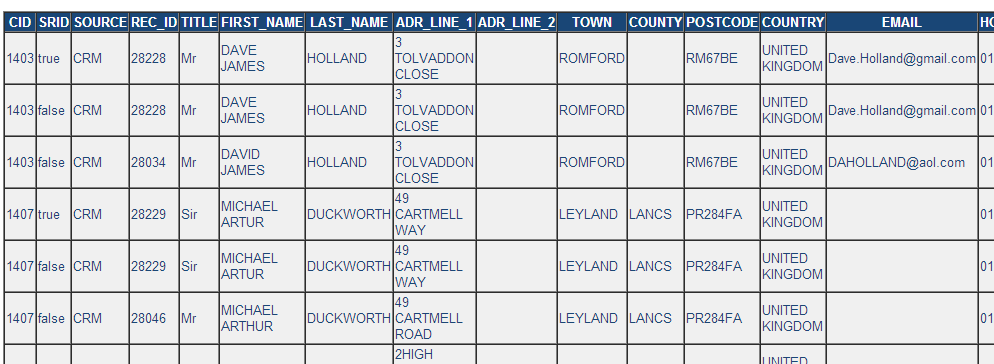
* + 1. Use Cluster Node and SRI Node to check if cluster result is as expected

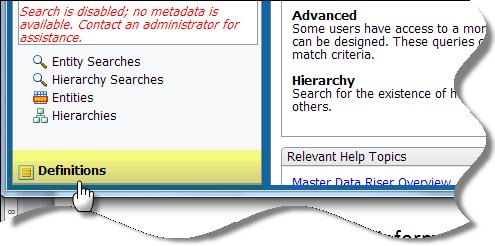
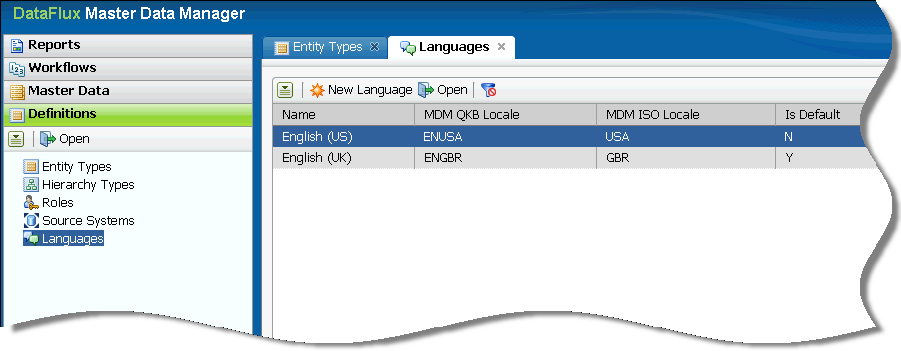
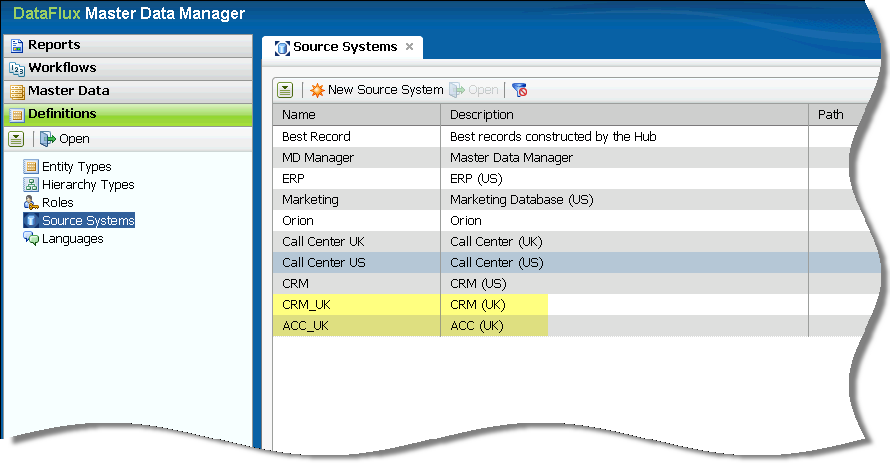


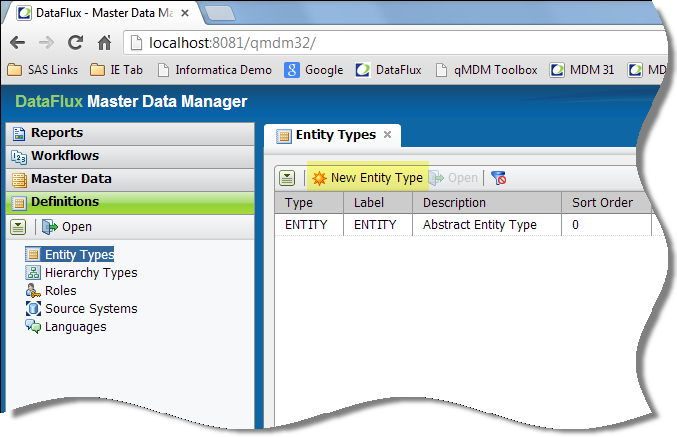
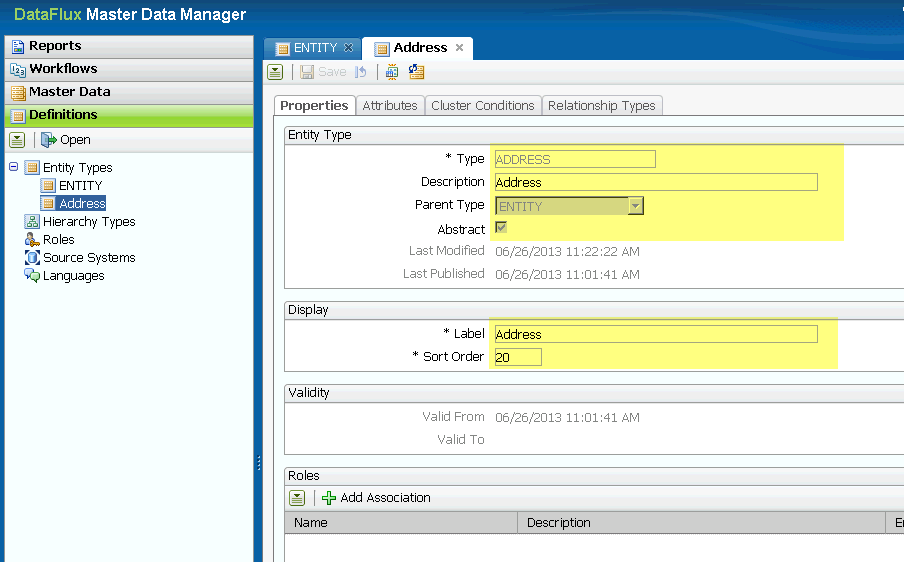




* + 1. Output result to html report for example.   
       This will help to see if the result is as expected. Adjust standardization and Matching until you reach a satisfying result.



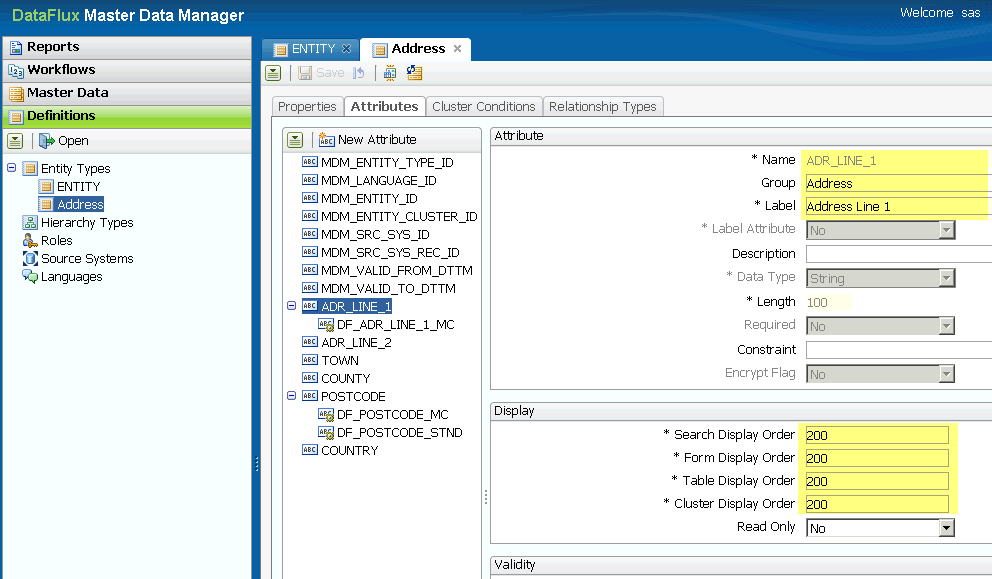
1. **Create Entity in qMDM** (using MD Manager)
   1. Log in to MD Manager and go to Definitions  
      
   2. Register new Language
      1. **Name:** English (UK)
      2. **QKB Locale:** ENGBR
      3. **ISO Locale:** GBR
      4. **Set as Default.** (Set the UK locale to Y and US locale to N)  
         
   3. Register new Source Systems
      1. CRM
      2. ACC  
         

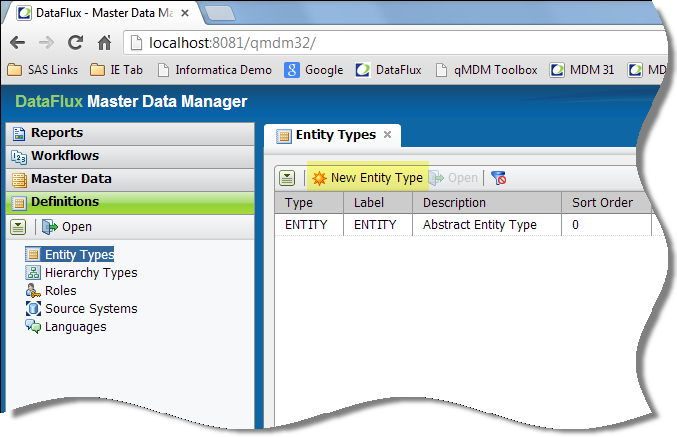
* 1. Create New Entity Types.
     1. Click on Entity Types =>New Entity Type to create **Entity ADDRESS**  
          
        
     2. In **Properties** set:
        1. Entity Name: ADDRESS
        2. Description: Address
        3. Abstract: Tick Box
        4. Label: Address
        5. Sort Order: 20  
           
     3. Use information from 2.4 and 2.5 to create the Entity Attributes and Flux Attributes

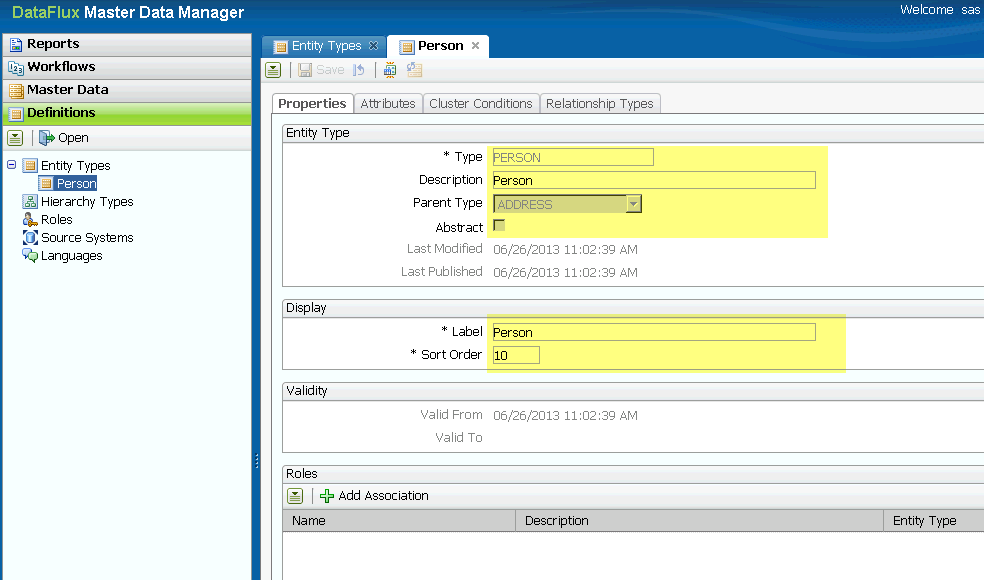
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attribute** | **Type** | **Length** | **MC** | **STND** | [**Display Order**](#MD_Manager_Display_Order) | **Group** |
| Address | ADR\_LINE\_1 | String | 100 | X |  | 200 | Address |
| Address | ADR\_LINE\_2 | String | 100 |  |  | 210 | Address |
| Address | TOWN | String | 50 |  |  | 220 | Address |
| Address | COUNTY | String | 30 |  |  | 230 | Address |
| Address | POSTCODE | String | 10 | X | X | 240 | Address |
| Address | COUNTRY | String | 30 |  |  | 250 | Address |

See also file: [Entity\_Design.xls](file:///D:\qMDM_Workshop\Docs\Entity_Design.xlsx)

* + 1. In Tab *Attributes* set Attributes, Flux Attributes and Display Order.
       - * Use information from the table above to create the Attributes, Display Order and Display Groups



* + 1. Save Entity
    2. Click on Entity Types =>New Entity Type to create **Entity PERSON**  
       
    3. In **Properties** set:
       1. Entity Name: PERSON
       2. Description: Person Entity
       3. Parent Type: ADDRESS
       4. Label: Person
       5. Sort Order: 10

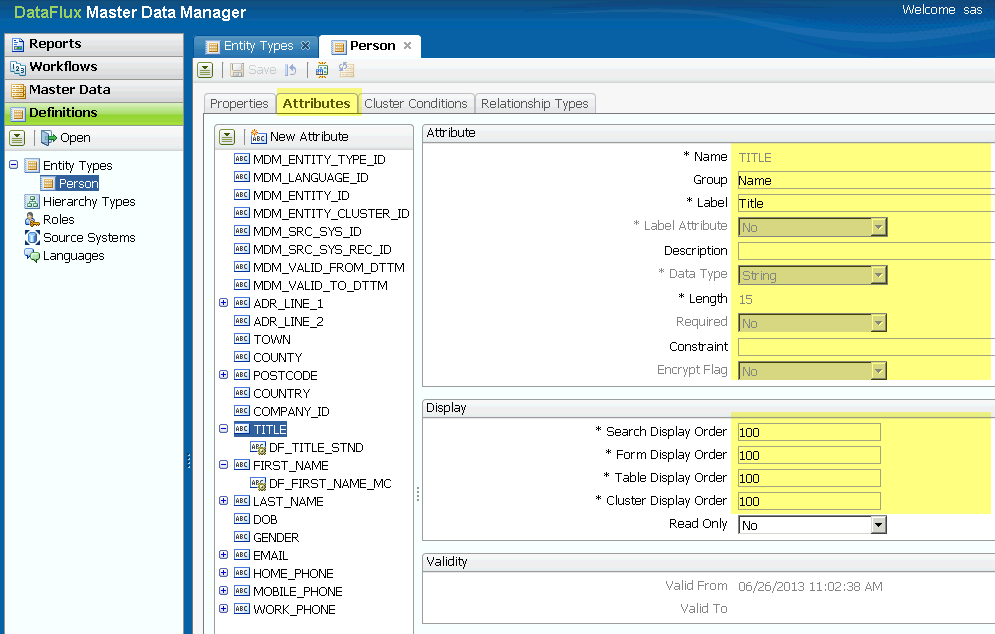


* + 1. Use information from 2.4 and 2.5 to create the Entity Attributes and Flux Attributes.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attribute** | **Type** | **Length** | **MC** | **STND** | [**Display Order**](#MD_Manager_Display_Order) | **Group** |
| Person | TITLE | String | 15 |  | X | 100 | Name |
| Person | FIRST\_NAME | String | 50 | X |  | 110 | Name |
| Person | LAST\_NAME | String | 50 | X |  | 120 | Name |
| Person | DOB | Date |  |  |  | 130 | Name |
| Person | GENDER | String | 2 |  |  | 140 | Name |
| Person | EMAIL | String | 100 |  | X | 300 | Contact |
| Person | HOME\_PHONE | String | 20 |  | X | 310 | Contact |
| Person | MOBILE\_PHONE | String | 20 |  | X | 320 | Contact |
| Person | WORK\_PHONE | String | 20 |  | X | 330 | Contact |
| Person | COMPANY\_ID | String | 5 |  |  | 0 | Meta |

See also file: [Entity\_Design.xls](file:///D:\qMDM_Workshop\Docs\Entity_Design.xlsx)

* + 1. In Tab Attributes set Attributes, Flux Attributes and Display Order.
       - * Use information from the table above to create the Attributes, Display Order and Display Groups
         * For LAST\_NAME set Label Attribute to **Yes**

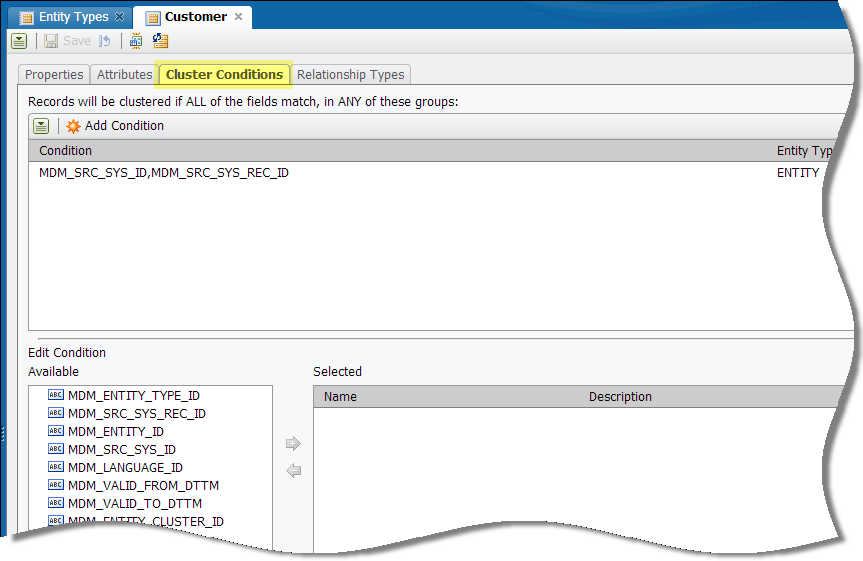


**Note:** Don’t create a field for the **REC\_ ID** as it will be mapped to the default Attribute **SRC\_SYS\_REC\_ID** in the load job.

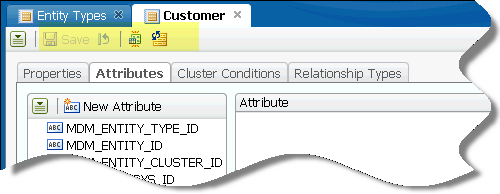
* + 1. Match Rules.

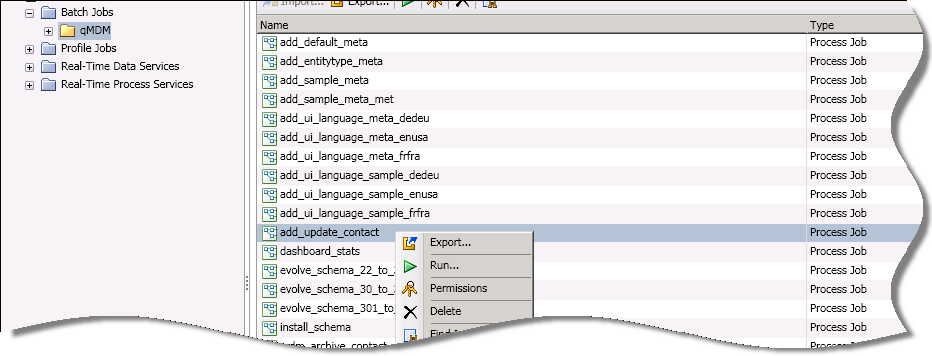
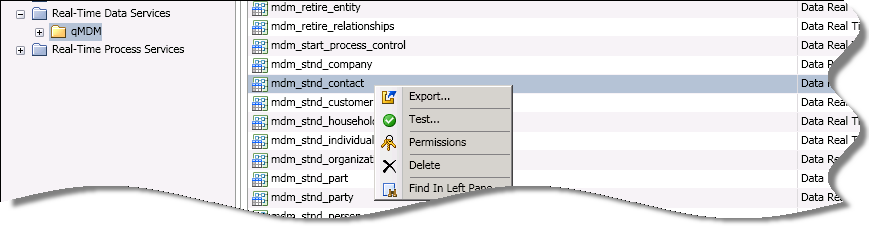
Go to Cluster Conditions to set the Match Rules for the Entity Definition. Use the Rules defined in 2.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | C1 | C2 | C4 | C5 | C6 |
| DF\_FIRST\_NAME\_MC | X |  | X | X | X |
| DF\_LAST\_NAME\_MC |  | X | X | X | X |
| DF\_ADR\_LINE\_1\_MC | X | X |  |  |  |
| POSTCODE\_MC | X | X | X | X | X |
| DATE\_OF\_BIRTH | X | X |  |  | X |
| DF\_HOME\_PHONE\_MC |  |  |  |  |  |
| DF\_MOBILE\_PHONE\_MC |  |  | X |  |  |
| DF\_EMAIL\_STND |  |  |  | X |  |

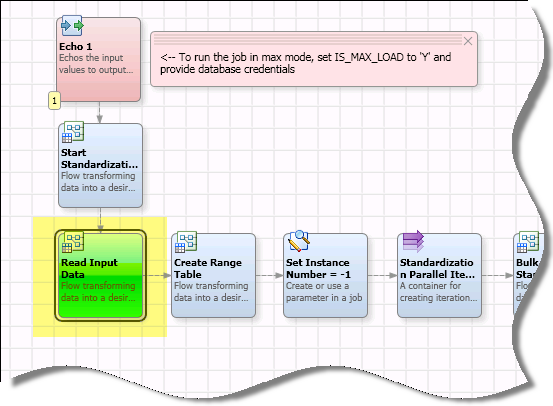
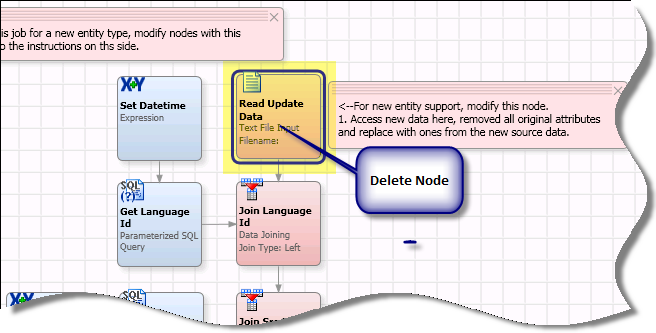


* + 1. Save Entity
    2. Publish Entity ADDRESS
    3. Publish Entity PERSON
    4. Generate Jobs. If prompted tick box to generate all Jobs.



1. **Configure generated jobs**
   1. In DM Studio export entity jobs for Address and Person from DM Server into the Repository where qMDM is installed
      1. DM Server\Batch Jobs\qMDM\**add\_update\_<Entity\_Name>** => Repository\batch\_jobs\qMDM  
         (For Entity Person only)  
         
      2. DM Server\ Real-Time Data Services \qMDM\**mdm\_stnd\_<Entity\_Name>** => Repository\data\_services\qMDM  
         (For both Entities Person and Address)  
         

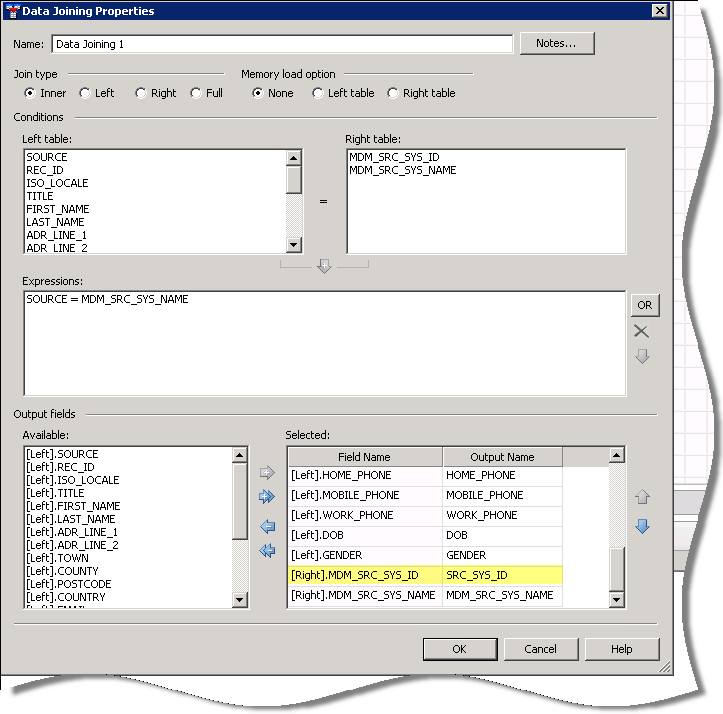
**Hint:** If DM Studio and DM Server are on the same machine it might be useful to have command line scripts to copy all entity jobs from the Server to the Client and vice versa. This will save time instead of each time manually exporting and importing the jobs.

* 1. Modify **add\_update\_<Entity\_Name>   
     Tip:** When changing or adding a Node mark those in a different colour so you know that these have been added or changed.
     1. Modify Node: Read Input Data  
        Double Click Node to edit contents  
        
     2. Delete Node: Read Update Data  
        
     3. Add ***Data Source Node*** to read PoC data from Staging Database
     4. Add ***SQL Query Node*** to receive SRC\_SYS\_ID.
        1. Data Source: MDM32
        2. Query:

SELECT MDM\_SRC\_SYS\_ID, MDM\_SRC\_SYS\_NAME

FROM MDM\_SRC\_SYS

WHERE MDM\_VALID\_TO\_DTTM is null

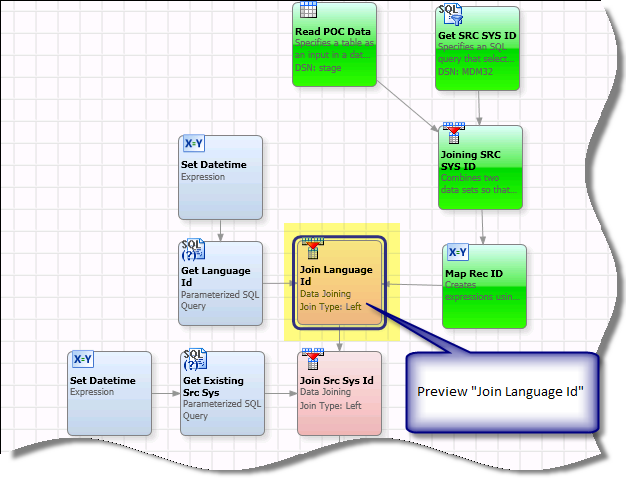
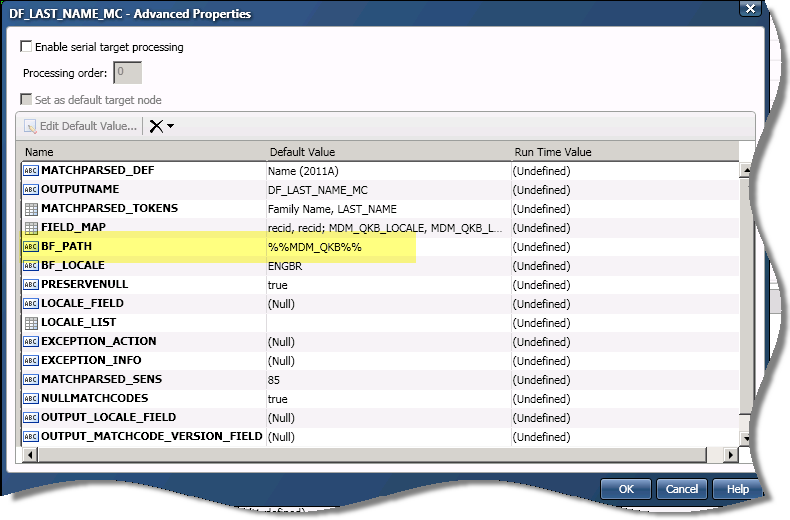
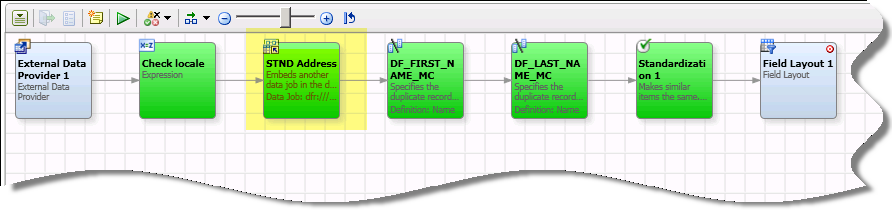
* + 1. Add **Data Join Node**
       1. Join SOURCE = MDM\_SRC\_SYS\_NAME
       2. Pass through all fields
       3. Rename MDM\_SRC\_SYS\_ID to SRC\_SYS\_ID  
          
    2. Map REC\_ID

As REC\_ID is an Integer we need to map it to the string variable SRC\_SYS\_ID

* + - 1. Add ***Expression Node*** to set SRC\_SYS\_REC\_ID
      2. In Expression Tab:

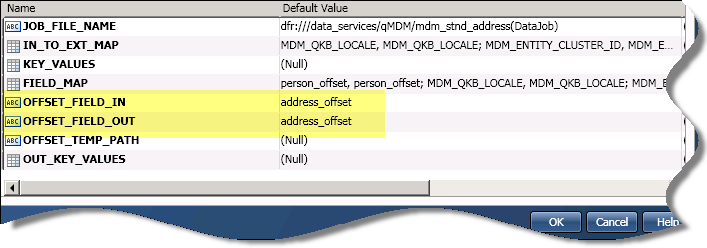
string SRC\_SYS\_REC\_ID

SRC\_SYS\_REC\_ID= REC\_ID

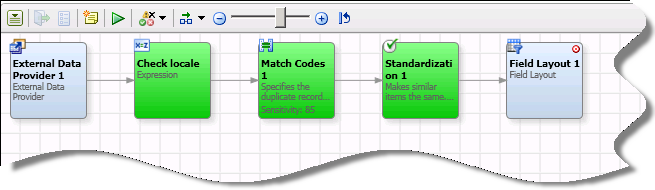
* + - 1. Preview on Node: Join Language ID   
          to check that all required fields are passed through. If an error occurs fix as appropriate.   
         
      2. Save Job
  1. Modify **mdm\_stnd\_<Entity\_Name>**See also file: [Entity\_Design.xls](file:///D:\qMDM_Workshop\Docs\Entity_Design.xlsx) as reference for fields to be generated. **We need to modify the stnd jobs for both entities, Person and Address.**
     1. Open the job **mdm\_stnd\_<Entity\_Name>** in Repository\data\_services\qMDM and add all nodes to the job necessary for the required Match Codes and Standardization.
     2. The default field names for the Flux Attributes have this format:  
        Match Code: DF\_<Field\_Name>\_MC  
        Standardized: DF\_<Field\_Name>\_STND
     3. Change Node **Check locale** to check on ENGBR as this is the locale being used for this POC. If more than one locale is used you have to add it as required.  
        if (not (`MDM\_QKB\_LOCALE`=='ENGBR'))
     4. On all Nodes where the QKB is called set QKB in Advanced settings to   
        BF\_PATH= %%MDM\_QKB%%  
        
     5. Add node Data Job (reference) to the stnd job and point it to the address stnd job  
          
        In node Check locale in section Expression add an integer for the reference job offset variable:  
        integer address\_offset

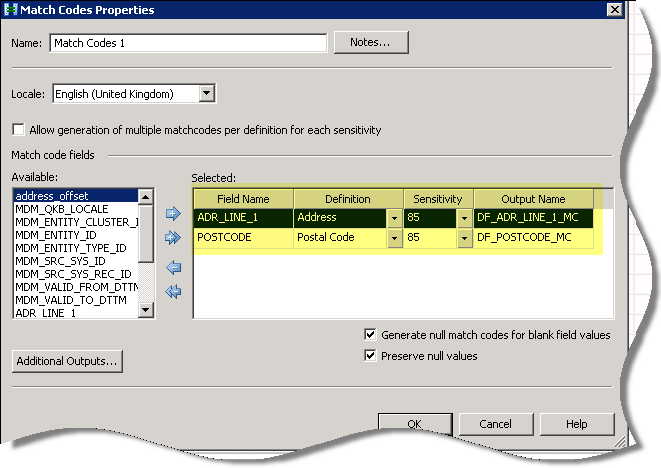
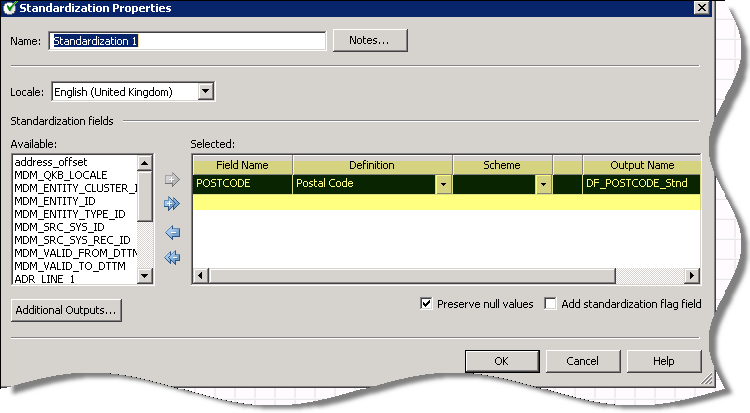
address\_offset= person\_offset

!!This is important: We cannot reuse the same offset field when calling other embedded jobs!!!  
Go to STND Address Advanced settings and set the offset fields to the just created offset field address\_offset:  
OFFSET\_FIELD\_IN= address\_offset  
OFFSET\_FIELD\_OUT= address\_offset



* + 1. Open job stnd\_address (via stnd\_person)
    2. Change Node **Check locale** to check on ENGBR as this is the locale being used for this POC. If more than one locale is used you have to add it as required.  
       if (not (`MDM\_QKB\_LOCALE`=='ENGBR'))
    3. On all Nodes where the QKB is called set QKB in Advanced settings to   
       BF\_PATH= %%MDM\_QKB%%
    4. Copy across appropriate nodes from stnd\_test job to generate Flux Attributes for Address.



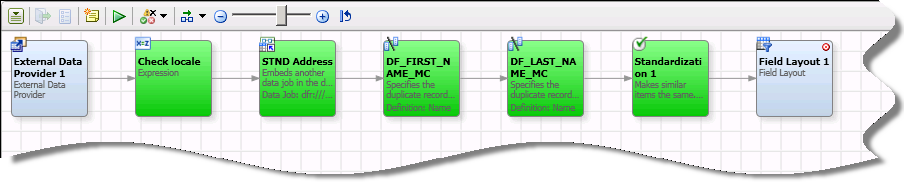
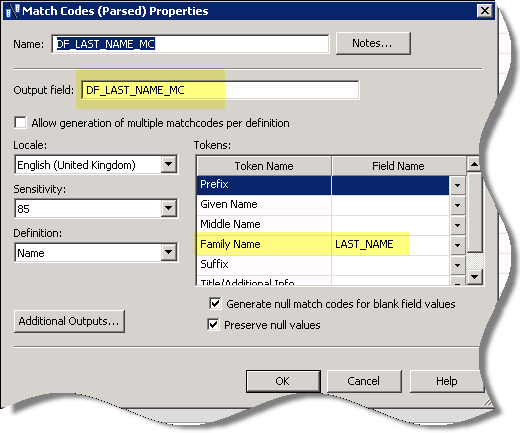
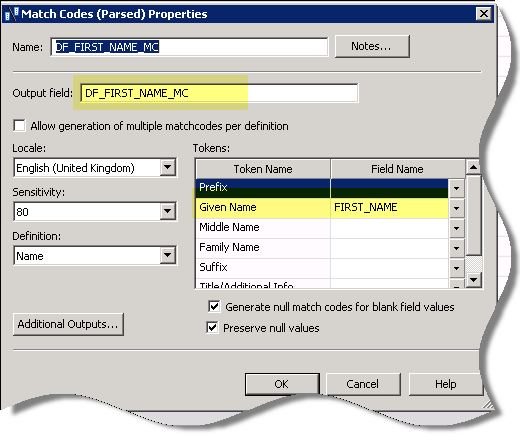
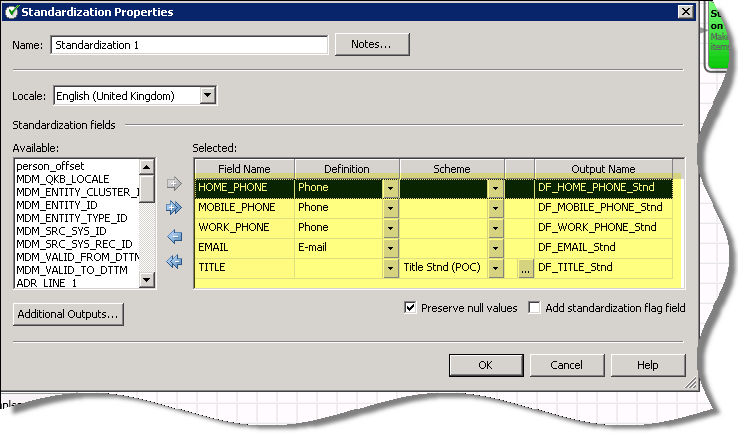
  


* + 1. Make sure all fields are passed through correctly by previewing the job.

Preview Node: Field Layout 1.

If the error occurs: *Error in Check locale: Expression Plugin - Unsupported locale: abcdefghij.*This indicates that all required Flux Attribute fields are covered in the job. Otherwise fix things are appropriate.

* + 1. Save job stnd\_address
    2. Close Job stnd\_address
    3. In stnd\_person copy across appropriate nodes from stnd\_test job to generate Flux Attributes for Person.

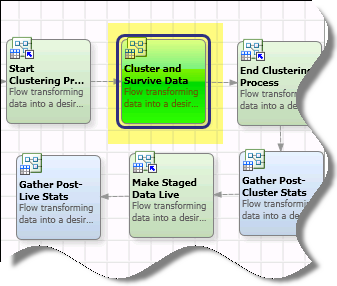
* + 1. Make sure all fields are passed through correctly by previewing the job.

Preview Node: Field Layout 1.

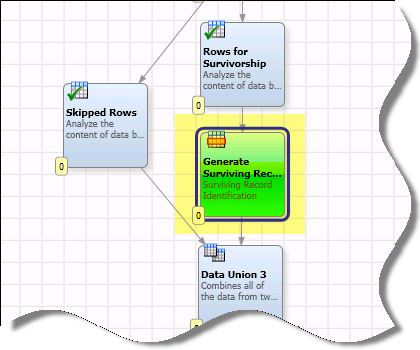
If the error occurs: *Error in Check locale: Expression Plugin - Unsupported locale: abcdefghij.*This indicates that all required Flux Attribute fields are covered in the job. Otherwise fix things are appropriate.

* + 1. Save job stnd\_person
    2. Close Job stnd\_person

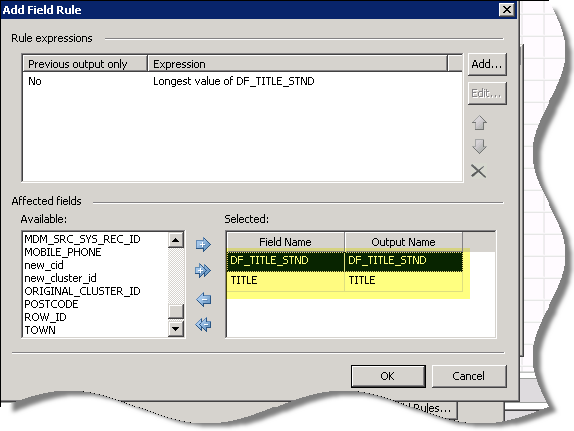
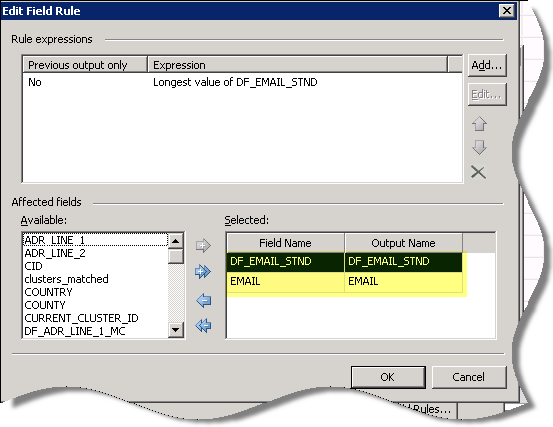
**Note:** Do not change the first or the last node in a standardization job (stnd\_<Entity>) as this will cause problems in other generated entity job!

* 1. **Set Survivorship Rules**
     1. Open Node: Cluster and Survive Data in Job **add\_update\_<Entity\_Name>**  
        

Go to Note: ***Generate Surviving Record*** and apply Field Rules

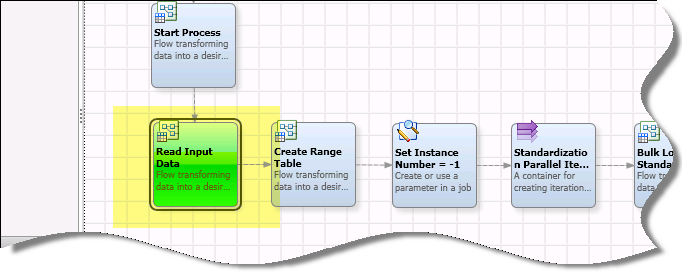
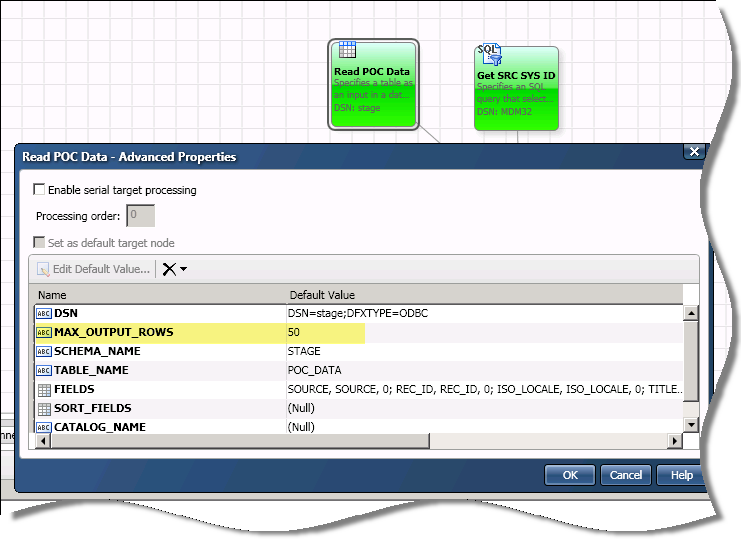


|  |  |  |
| --- | --- | --- |
| **Field Rule** |  |  |
| DF\_EMAIL\_STND=longest value |  |  |
| **Affected Fields** | **Field Name** | **Output Name** |
|  | DF\_EMAIL\_STND | DF\_EMAIL\_STND |
| **Field Rule** |  |  |
| DF\_TITLE\_STND=longest value |  |  |
| **Affected Fields** | **Field Name** | **Output Name** |
|  | DF\_TITLE\_STND | DF\_TITLE\_STND |



**Note:** To keep the standardized field in sync with its original field we need to copy both the standardized and original filed value

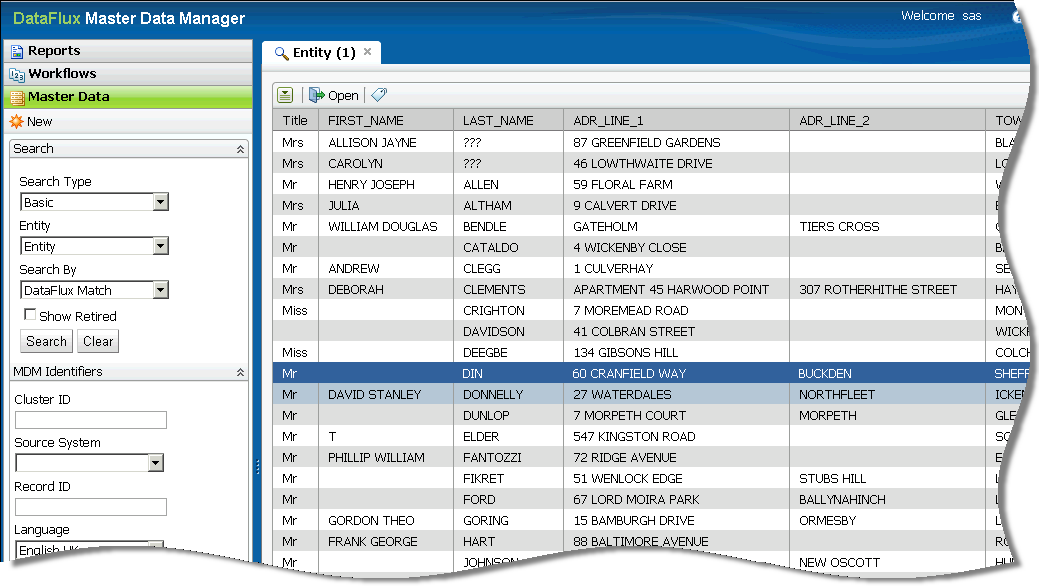
* + 1. Save Job

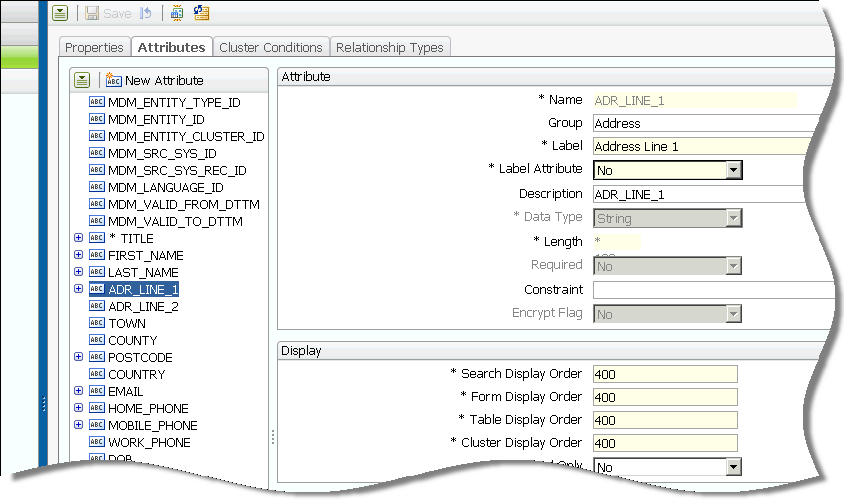
1. **Run Job on small sample** (to make sure all changes are correct and working)
   1. In job **add\_update\_<Entity\_Name>** open node: **Read Input Data**  
      
   2. Set MAX\_OUTPUT\_ROWS to 50 in Advanced Properties on node: Read POC Data.  
      
   3. Run Job
   4. If Job is not successful fix as appropriate
2. **Set SRI Rules for MD Manager Jobs**

MD Manager uses some Entity jobs where the SRI and Field Rules need to be set if used in the load Jobs.

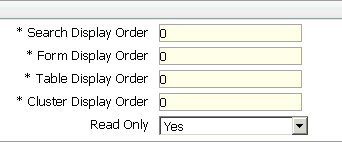
* 1. In DM Studio export jobs from DM Server into the Repository where qMDM is installed
     1. DM Server\ Real-Time Data Services \qMDM\ **mrm\_add\_<Entity\_Name>** => Repository\data\_services\qMDM  
        DM Server\ Real-Time Data Services \qMDM\ **mrm\_move\_<Entity\_Name>\_records** => Repository\data\_services\qMDM  
        DM Server\ Real-Time Data Services \qMDM\ **mrm\_revert\_move\_<Entity\_Name>** => Repository\data\_services\qMDM
     2. Open each job, go to Node: ***Generate Surviving Record*** and set Field Rules.  
        See **4.4** **Set Survivorship Rules** and follow these steps to set the appropriate rules.

1. **Move MD Manager Jobs back to qMDM Server**
   1. In DM Studio go to DM Server and import all modified Real Time jobs to DM Server:
      1. **mdm\_stnd\_<Entity\_Name>  
         mrm\_add \_<Entity\_Name>  
         mrm\_move \_<Entity\_Name>\_records  
         mrm\_revert\_move \_<Entity\_Name>**
      2. Repository\data\_services\qMDM\ **<Modified\_Job>**   
         => DM Server\ Real-Time Data Services \qMDM\
2. **Check result in MD Manager**

Login to MD Manager and check the loaded data  


1. **Run Job on full PoC data**Remove the setting (50) from MAX\_OUTPUT\_ROWS set in 6.2.   
   Set MAX\_OUTPUT\_ROWS to 0  
   Run Job
2. **Change MD Manager Screens**
   1. When the result is correct and the hub doesn’t need to get re-build again, change the Screen Layout for MD Manager
   2. Open Entity Definition in MD Manager
   3. Go to View Attributes Change values in section Display   
      

**Display Order Sections in MD Manager**



|  |  |
| --- | --- |
|  |  |

1. **Some Useful Tips**
   1. Tip => Change Label for default Attributes



* 1. Tip => Suppress technical Fields in MD Manager

Some Attributes like MDM\_ENTITY\_ID are very technical and there might not be much sense in presenting them to the user. It might be useful not to show these attributes on the screen.