Oracle® Data Profiling and Oracle Data Quality for Data Integrator

Sample Tutorial
11g Release 1 (11.1.1.3)

January 2011

Oracle® Data Profiling and Oracle Data Quality for Data Integrator Sample Tutorial, 11g Release 1 (11.1.1.3)

Table of Contents

Introduction to Oracle Data Quality Products

Oracle Data Quality Products

Oracle Data Quality products - **Oracle Data Profiling** and **Oracle Data Quality for Data Integrator** - extend the inline Data Quality features of **Oracle Data Integrator** to provide more advanced data governance capabilities.

Oracle Data Profiling is a data investigation and quality monitoring tool. It allows business users to assess the quality of their data through metrics, to discover or infer rules based on this data, and to monitor the evolution of data quality over time.

Oracle Data Quality for Data Integrator is a comprehensive award-winning data quality platform that covers even the most complex data quality needs. Its powerful rule-based engine and its robust and scalable architecture places data quality and name & address cleansing at the heart of an enterprise data integration strategy.

Tutorial Contents

This tutorial guides you through a first project involving data profiling and data quality.

You will first start by configuring a new installation of the Oracle Data Quality products in order to run projects.

The Oracle Data Profiling Tutorial section will guide you through an investigation process on a set of files to detect data anomalies and inconsistencies, and create new business rules on this data.

Finally, the Oracle Data Quality for Data Integrator Tutorial section will show you how to create a data quality process to cleanse a file containing incorrect and incomplete name and address records.

Recommended Readings

It is recommended that you first read the *Oracle Data Quality for Data Integrator - Getting Started Guide* to have an overview of the user interface, the key concepts and steps for data profiling and quality.

Prepare for the Tutorial

Install Oracle Data Quality and Data Profiling

Refer to the *Oracle Data Integrator Installation Guide* for installing Oracle Data Quality products as well as Oracle Data Integrator.

Setup the Data Files

1. On your server, create a directory where the sample files will be stored. We will refer to this directory as **ODQ_SAMPLE_FILES** throughout this document. (for example `C:\demo\oracledq`).
2. Copy and unzip the file named *oracledq-sample-data-134552.zip* to the **ODQ_SAMPLE_FILES** directory.

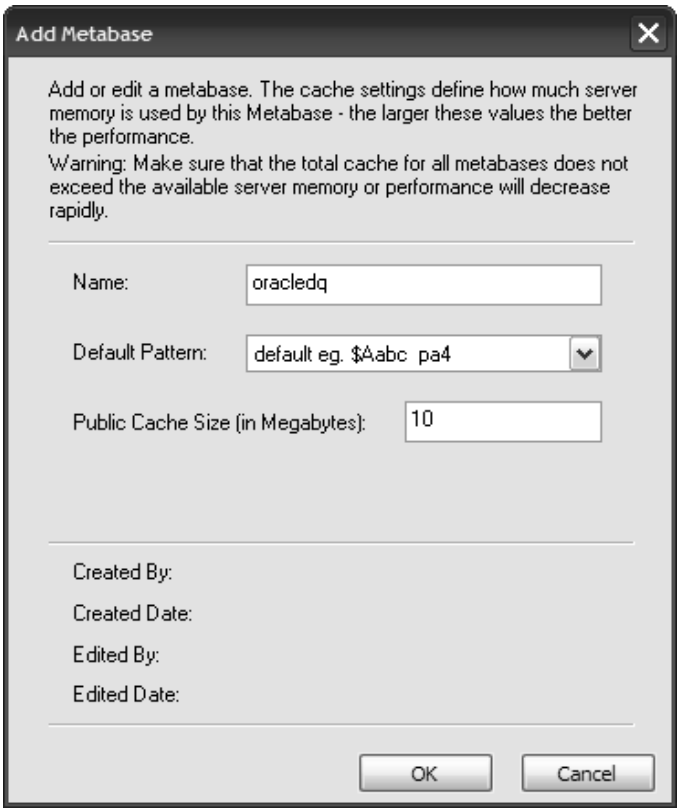
Install the Postal Directories

1. Extract the *oracledq_sample_directory.zip* to a temporary directory on your file system.
2. Copy the content of this temporary directory into the Oracle Data Quality server directory, in the `tables\postal_tables\` sub-directory (for example `C:\Oracle\product\11.1.1\odidq_1\oracledq\tables\postal_tables`). Overwrite existing files.

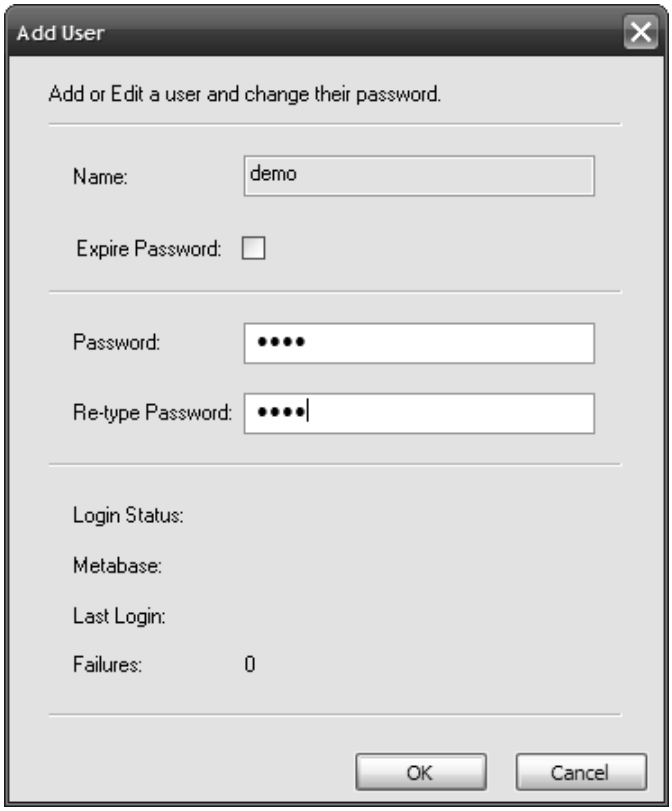
Note: The sample postal directory will allow enough coverage to get through the sample data. Only the specific locations useful for the sample data have been included, and not the entire country postal directory. This sample postal directory cannot be used with the Postal Directory Browser.

Configure the Metabase and the Connections

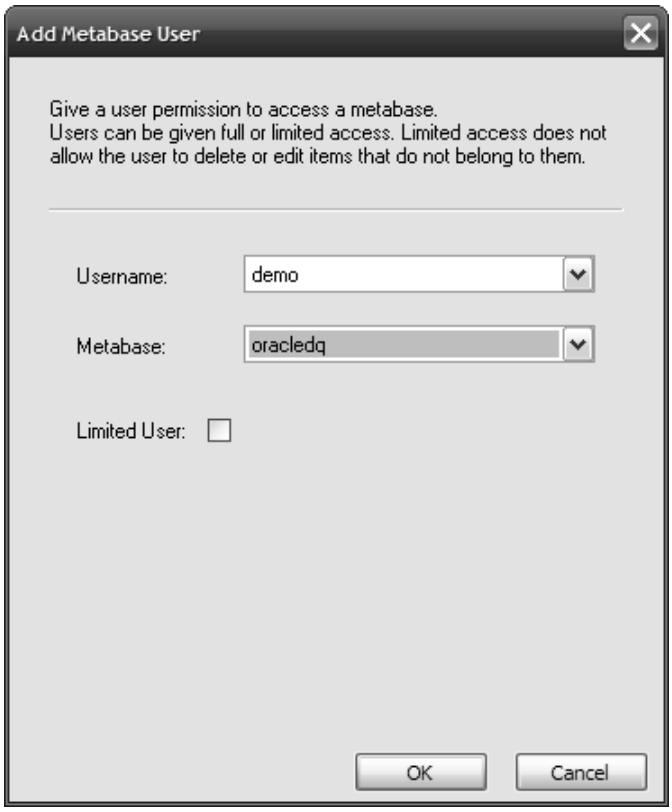
1. Make sure Oracle Data Quality and Data Profiling, as well as Oracle Data Integrator are installed and working.
2. Select **Start > All Programs > Oracle > Oracle Data Profiling and Quality > Metabase Manager** to Log in to the Metabase Manager as the Metabase Administrator (*madmin*)
3. Select **Tools > Add Metabase** from the menu
4. Add a metabase named *oracledq*, with the default pattern and a Public Cache Size of 10 Mb, and then click **OK**.



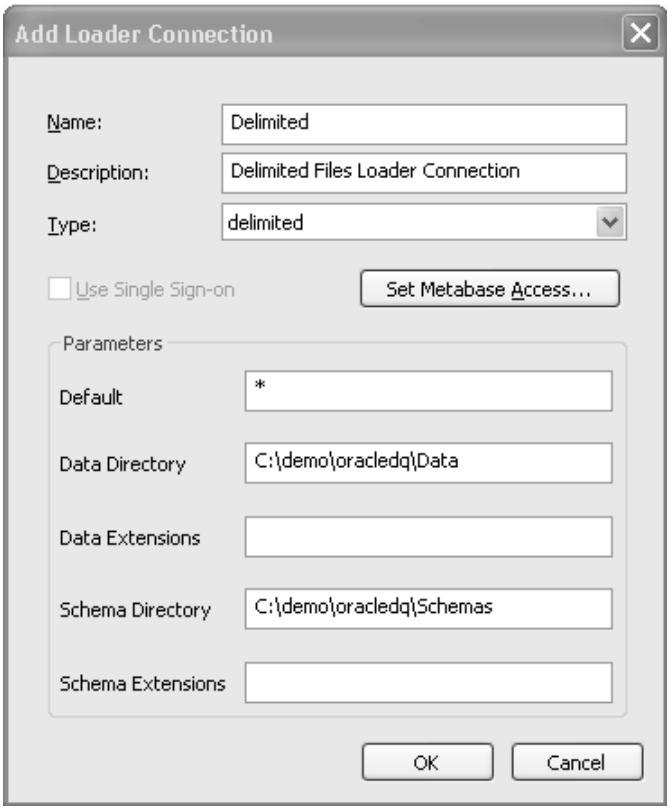
5. Select **Tools > Add User** from the menu
6. Add a User named *demo* with the password *demo*, as shown below, then click **OK**.



7. Select **Tools > Add Metabase User** to add the *demo* user to the *oracledq* metabase, as shown below, and then click **OK**.



8. Select **Tools > Add Loader Connection**.
   Create a loader connection for delimited files as shown below.
   - Name: `Delimited`
   - Description: `Delimited Files Loader Connection`
   - Type: `delimited`
   - Default filter: `*`
   - Data directory: **ODQ_SAMPLE_FILES**`\Data`
     (for example: C:\demo\oracledq\Data)
   - Schema directory: **ODQ_SAMPLE_FILES**`\Schemas`
     (for example: `C:\demo\oracledq\Schemas`)



9. Close Metabase Manager.

Preload the Metabase

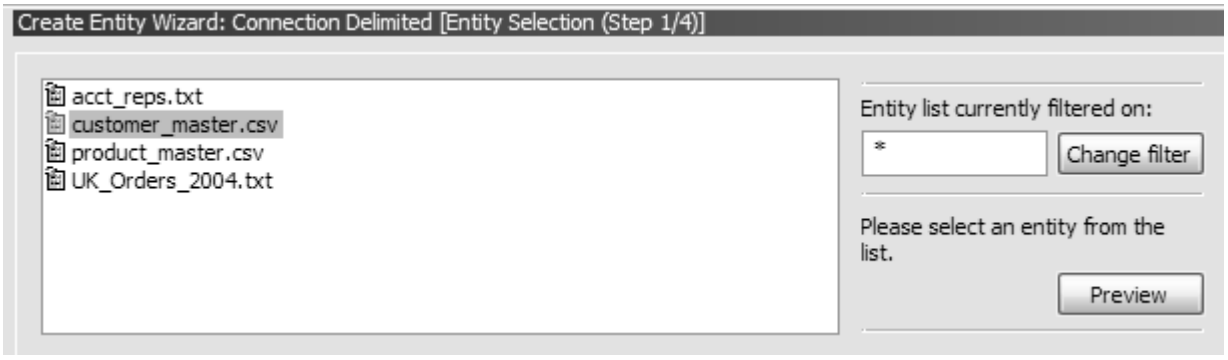
The Metabase contains both the description of the data structures as well as sample data to perform the Data Profiling operations and to design the Data Quality projects. The first step in the quality process is to preload the metabase.

In this sample, we will load the metabase with the flat files located in the **ODQ_SAMPLE_FILES** directory, using the delimited data loader defined in the previous chapter. Each of these files has a specific format that we will define when creating entities corresponding to these files.

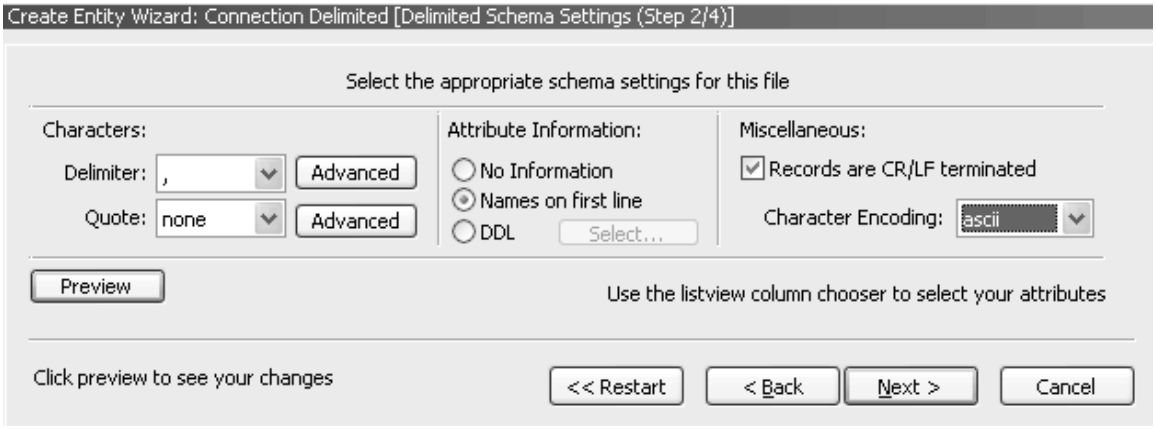
We first need to create an entity corresponding to the customer_master.csv source file with the following parameters:

Source File	File Info	Data Selection	Load Rows
customer_master.csv	delimiter: comma quote: none Names on first line CR/LF terminated: Y Character Encoding: ascii	Keep all data	All

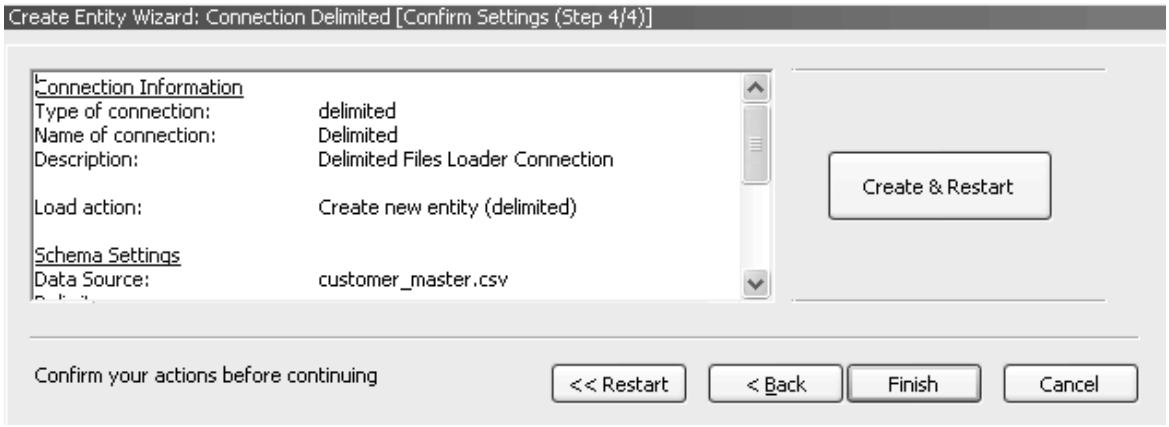
1. Login to Oracle Data Quality client (**Start > All Programs > Oracle > Oracle Data Profiling and Quality > Oracle Data Profiling and Quality**) using the following information:
   - Repository: primary
   - Metabase: oracledq
   - Username: demo
   - Password: demo
2. Select **Analysis > Create Entity** from the menu.
3. Select the *Delimited* Loader Connection, and then click **Next**.
4. Select the customer_master.csv file, and then click **Next**.



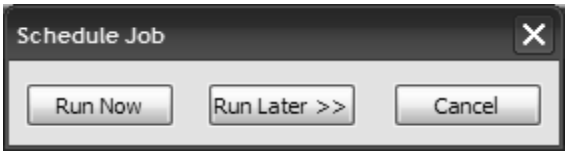
5. Set the file info as shown below, and then click **Next**.



6. Select **All Rows**, click **Next**, and then **Finish** in the next window.



7. Select **Run Now** in the Schedule job popup window.



8. Click on the Background Tasks icon in the toolbar ( ) to view the list of running task and wait until the job is complete.

Note: Remember to use this icon to review job completion every time you will start a job with the **Schedule Job** window.

9. Repeat the operation to create Entities using the following information:

Source File	File Info	Data Selection	Load Rows
product_master.csv	delimiter: comma quote: double Names on first line CR/LF terminated: Y Character Encoding: ASCII	Keep all data	All
acct_reps.txt	delimiter: <TAB>	Keep all data	All

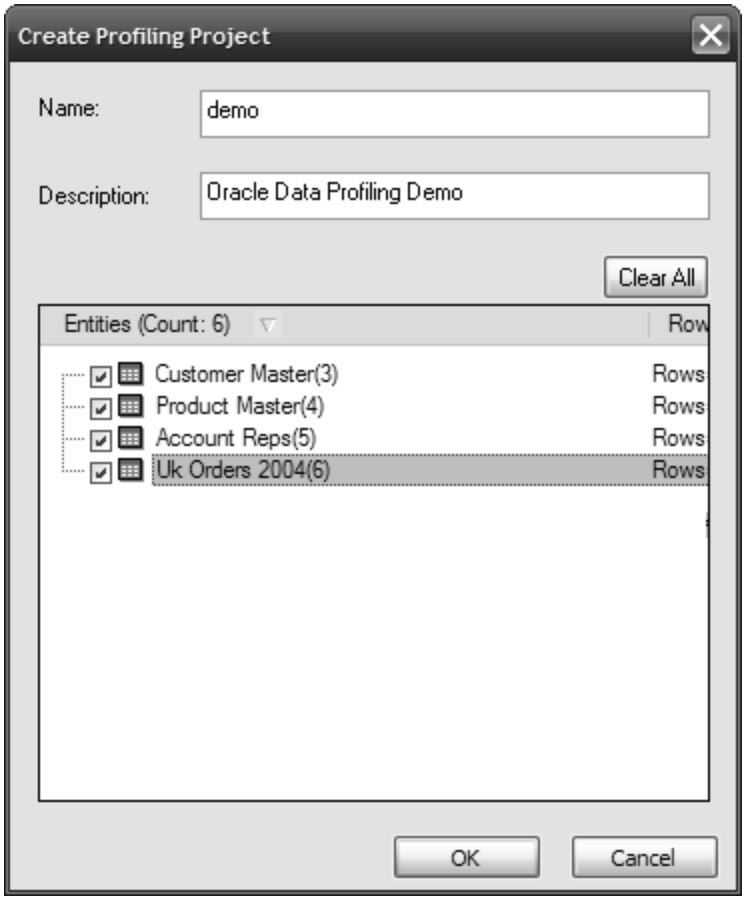
	quote: double DDL: acct_reps.ddl CR/LF terminated: Y Character Encoding: ASCII		
uk_orders_2004.txt	delimiter: tab quote: none Names on first line CR/LF terminated: Y Character Encoding: ASCII	Keep all data	All

All entities are created for the sources, and loaded with the source data. We can start profiling this data.

Investigate Data

This first profiling step will simply create a project with the four entities previously created. We will then explore one of these entities.

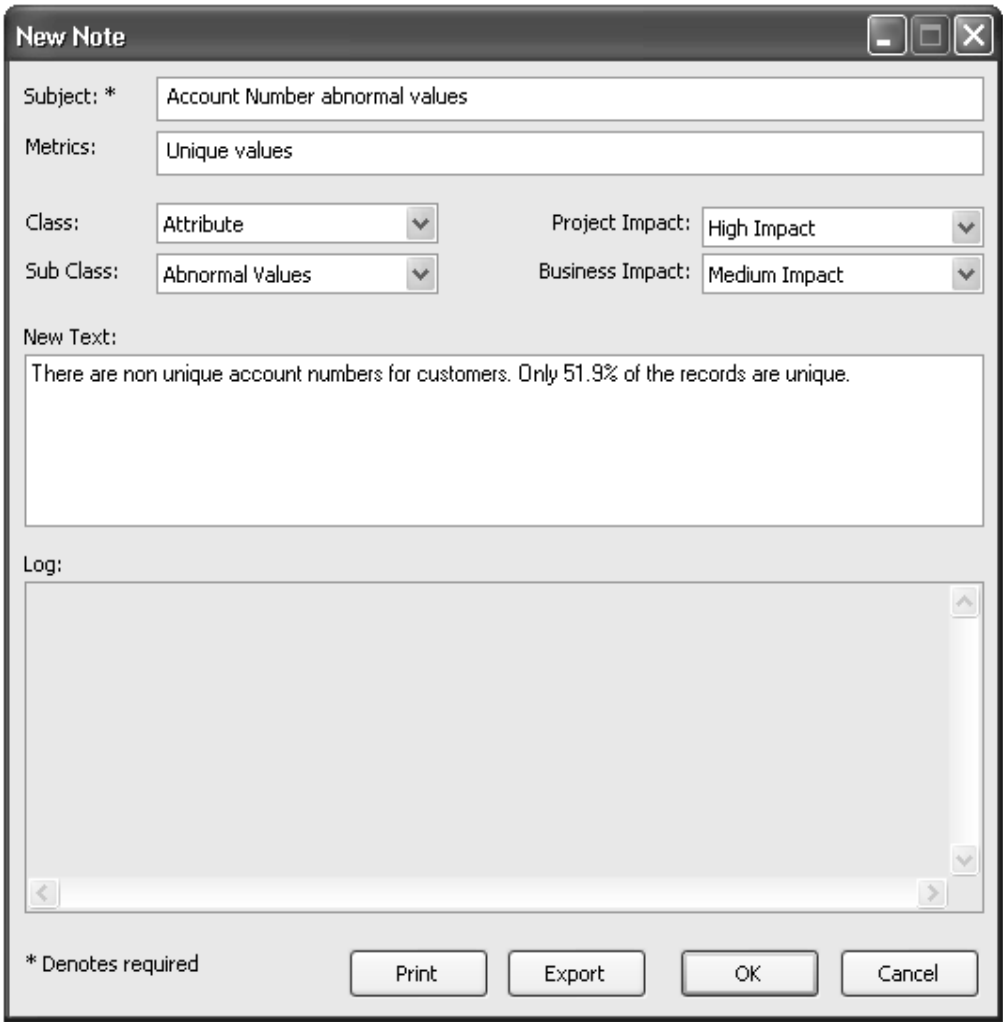
1. Create a Profiling Project named **demo**
    a. Select **Profiling** in the **Explorer**, right click and select **Create project…** in the popup menu.
    b. Enter the project name and description, then select all entities as shown below, then click **OK**.



2. Explore Entity level Metadata
    a. From the Profiling Project **demo**, expand **Customer Master** under the **Entities** folder
    b. Explore its **Metadata** folder and look at structural metadata such as:
        - Row min len – double-click to see the distribution for the shortest row found, then double click the distribution value to view the list of smallest rows.

- Row max len – Drill down as above to explore the longest rows.
- Source Type – type of the data source
- Data Source – name of the data source
- Load Sampling – sampling method (all rows)
- Entity Type
- Rows Loaded – double-click to see all rows for the Entity

3. Explore Attribute level Metadata
    a. Under **Customer Master**, expand the **Attributes** folder and double-click on the Attribute **Account Number**
    b. Examine Unique Values – notice that not 100% of the values are unique. Several customers exist with the same account number.
    c. Double-click on **Unique Values** to see the duplicate values.
    d. Drill down on a value (double click on a row in the table on the right panel) to see rows with similar account numbers
4. Add a note describing the discovered quality issue
    a. Right-click on the **Account Number** attribute and select **Notes > Add…**
    b. Enter the details for your note, as shown below, and then click **OK**.



5. Explore the different **Patterns** found for the **Phone** field.
    a. Under **Customer Master**, expand the **Attributes** folder and double-click on the attribute **Phone**
    b. Double-click on **Patterns**
    c. Drill down to pattern values with low frequencies.
    d. Drill down to the row level to see the rows with a given pattern.

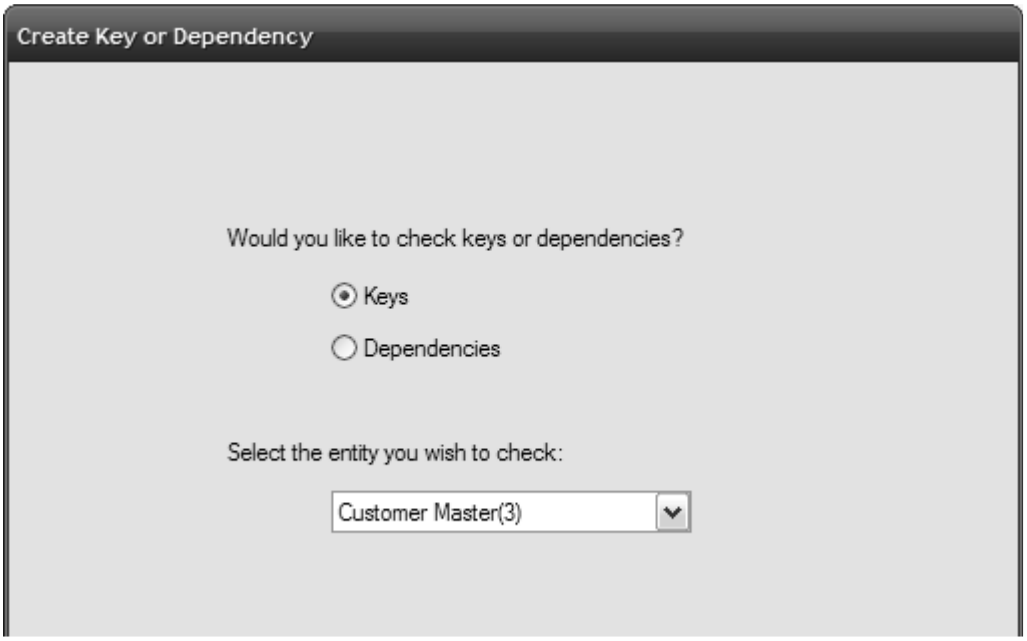
Explore Relationships within Entities

Oracle Data Profiling allows you to profile individual entities as well as relations between groups of entities. In this step, we will investigate possible keys for the Customer Master source data, and examine the dependencies between the customer account numbers and its references in the UK Orders file.

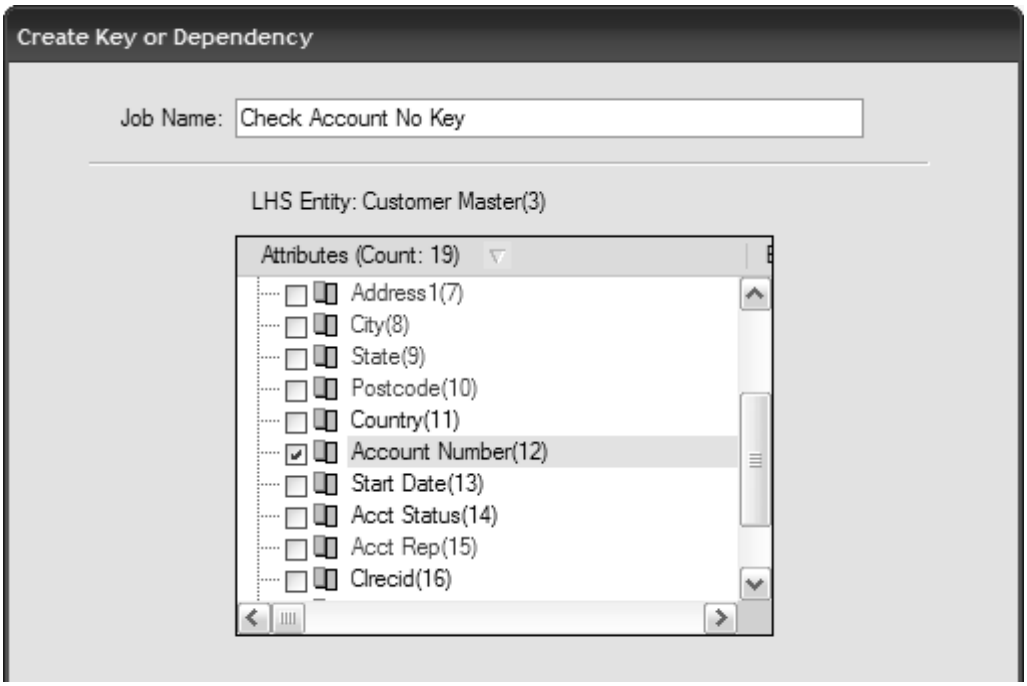
Explore Existing Keys and Find Alternate Keys

There is an implicit key defined on the Customer Master data source, the *Account Number*. We will now examine its validity as a key, and evaluate another column as a possible key;

1. From the Profiling Project **demo**, expand **Customer Master** under the **Entities** folder
2. Expand the **Metadata** node
3. Double-click on **Keys(Discovered)**
4. **Account Number** should be a key field in this data, but is not displayed in the list due to its low uniqueness.
5. Make **Account Number** a key through Create Key feature.
    a. Select **Analysis > Create Key or Dependency…** in the menu
    b. Select the *Customer Master* entity in the list then click on **Next**



    c. Name the Job “Check Account No Key”, then select the *Account Number* attribute in the list and then click **Finish**.



d. Click **Run Now** in the Schedule Job window.

6. Drill down to the rows with duplicate values.
    a. Double-click on **Keys(Discovered)** in the **Metadata** folder for **Customer Master**
    b. Double-click on the *Account Number* key in the table to drill down to the duplicate values
    c. Double-click on the values to drill down to the rows with duplicate values.
7. Identify **Clrecid** as a good alternate key.
    a. Double-click on **Keys(Discovered)** in the Metadata folder for **Customer Master**
    b. Double-click on the *Clrecid* key in the table to drill down to the duplicate values
    c. Double-click on the only duplicate value to drill down to the 3 rows using the same *Clrecid* value.

Examine Dependencies

There is a discovered dependency in the **Uk Orders 2004** table. An Order ID should have one and only one Account ID associated. We will examine now the potential conflicts on this dependency.

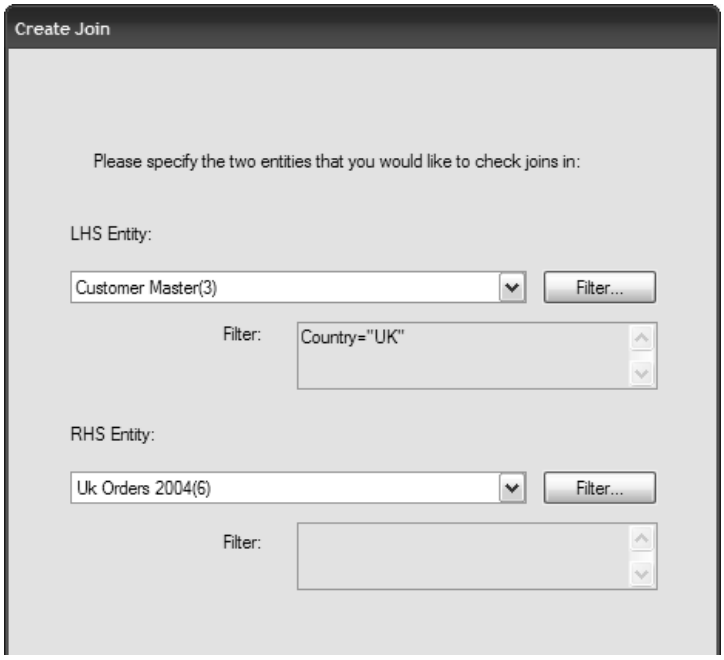
1. Double click the **Dependencies (Discovered)** node in the in the **Metadata** folder for **Uk Orders 2004**
2. Look at the dependency between **Order Id** and **Account Id**
3. Double-click on the row showing this dependency to drill down to see the two conflicts (several Accounts sharing one same Order)
4. Double-click on one of the rows showing a conflict instance to drill down to the rows with the conflicts

Explore Relationships between Entities (Joins)

1. Create a join between Customer Master and UK Orders 2004

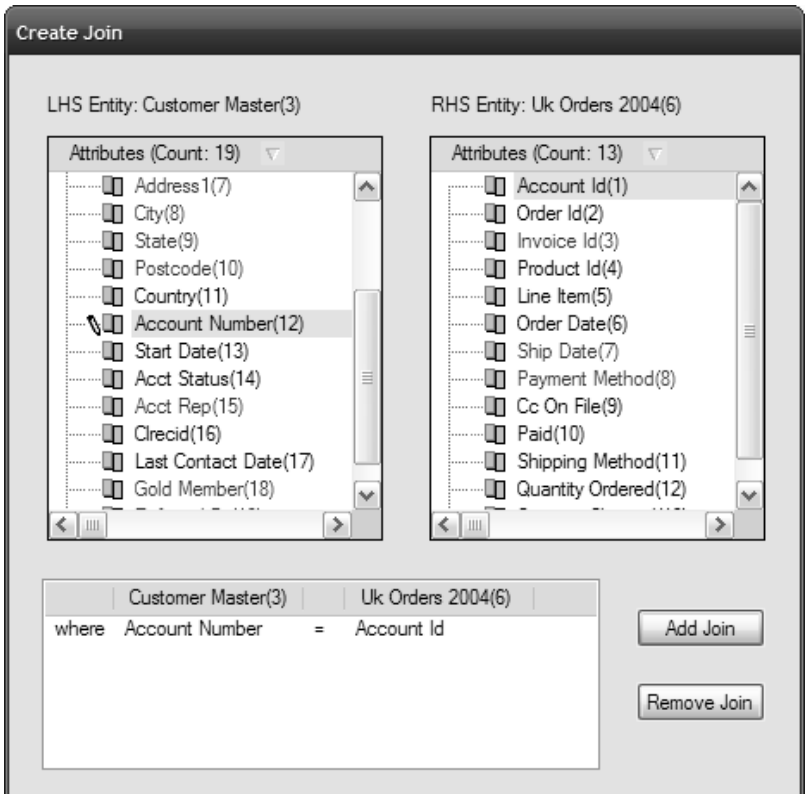
a. Select **Analysis > Create Join** in the menu.

b. Select the *Customer Master* and *UK Orders 2004* entities and then apply a filter to Customer Master.
Click on **Filter…** and enter `Country = "UK"` then click on Apply.



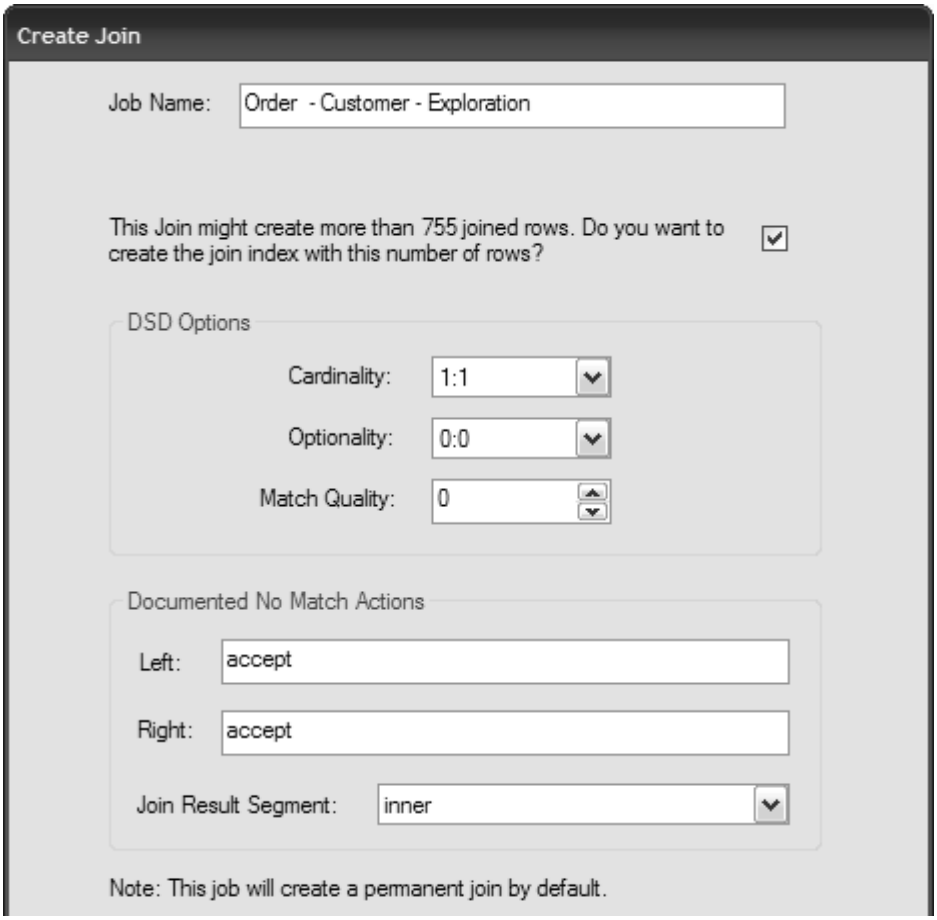
c. Click **Next**.

d. Join on **Account Number** and **Account Id**, by selecting these attributes under each entity and then clicking on the **Add Join** button.



e. Click **Next**.

f. Create the Join as shown below and then click **Finish**.



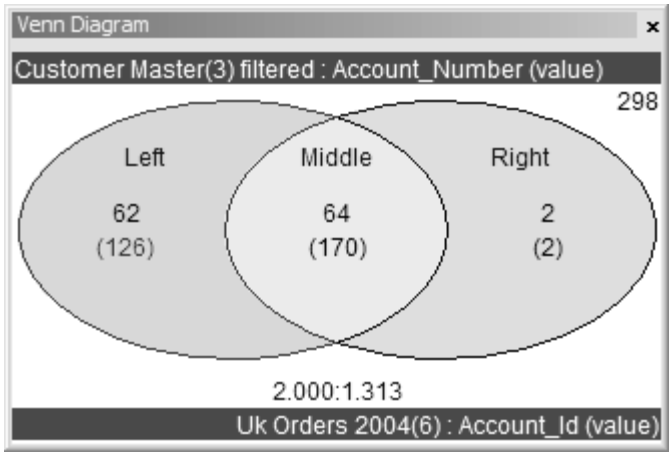
g. Click **Run Now** in the Schedule Job window.

h. Expand the **Permanent Joins** node under the *demo* project.

i. Double-click on the new join to display its properties in the right panel.

- Examine the number of matching and non-matching values.

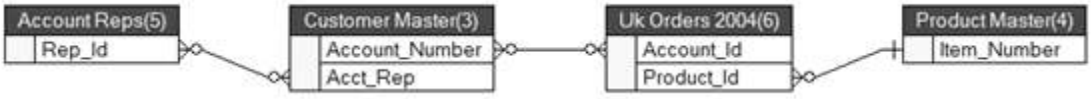
j. Right-Click on **Matching Values** in the list and then select **Venn Diagram**



k. Double click on sections of the diagram to:

- Drill down to customers without orders
- Drill down to orders that don't have an Account
- Drill down to customers that have orders

2. Reproduce the previous steps to create a join between **UK Orders 2004** and **Product Master**
    a. Join on **Product Id** and **Item Number**
    b. View Venn diagram
        - Drill down to orders without products
        - Drill down to products that haven't been ordered
        - Drill down to ordered products
3. Create a join between **Customer Master** and **Acct Reps**
    a. Join on **Acct Rep** and **Rep Id**
4. Right-click the **Permanent Join** node under the demo project, and then select **Entity Relationship Diagram**. The following diagram appears.



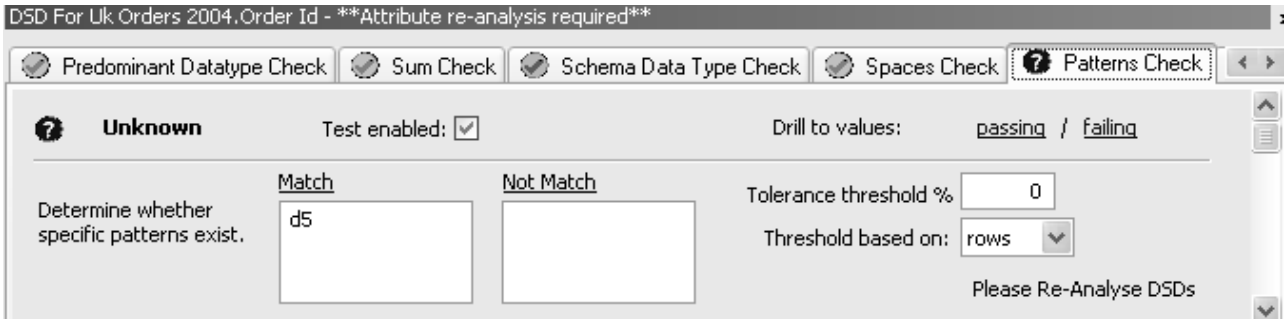
Check Data Compliance

In the profiling phase, you can check whether the data stored in the source files complies with a set of rules (based on patterns, values, data types, etc). These compliance checks allow you to evaluate the quality of each record.

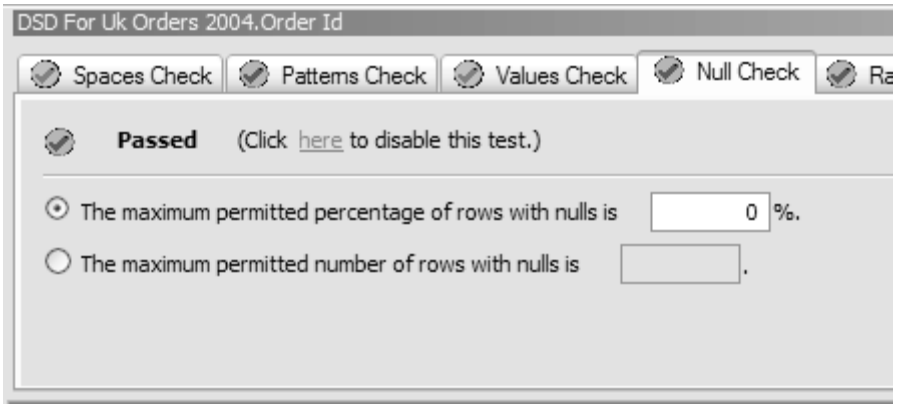
1. Add the following compliance checks to attributes in **Uk Orders 2004**

Attribute	DSD to apply
Order Id	Pattern Check - Pattern allowed d5 Null check – no null values allowed Acceptable values between 30560 and 32000
Payment Method	Valid Values are `CREDIT CARD`, `EFT`, `ACCOUNT` and `COD`

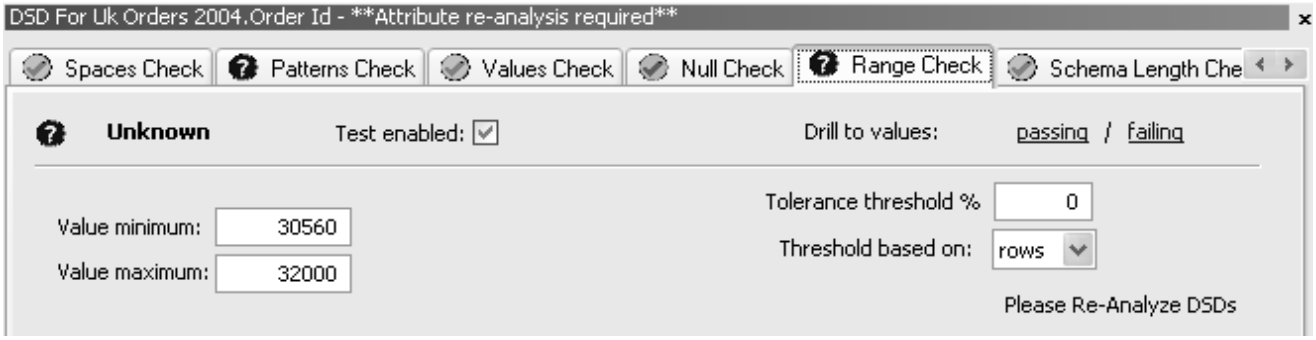
    a. Select in the *Demo* Project the **Entities > Uk Orders 2004 > Attributes > Order Id** attribute, right-click, then select **Edit DSD**.
    b. Select the **Patterns Check** tab, enable the test and then enter the `d5` pattern to match. Set the tolerance to 0% of rows.



c. Select the **Null Check** tab, enable the test, and make sure that no null row is allowed (0%).

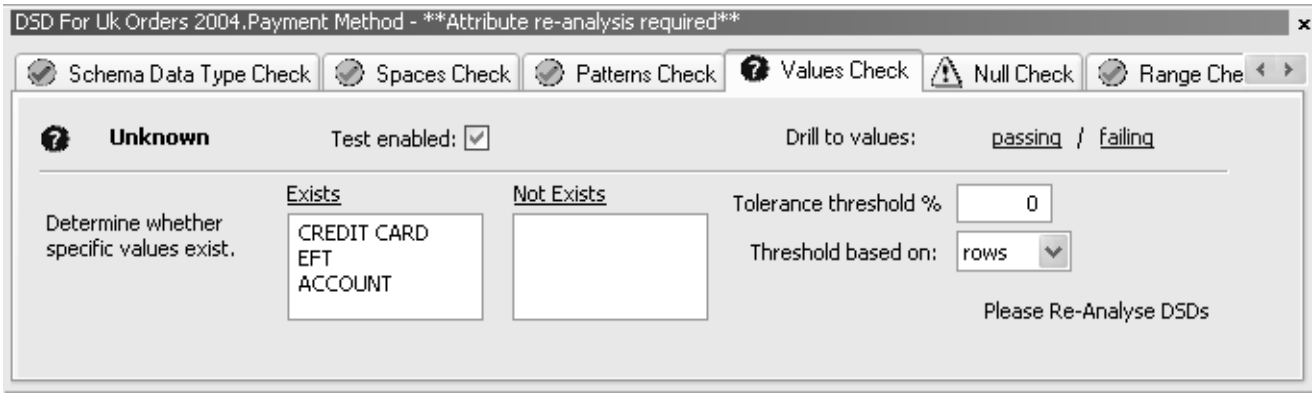


d. Select the **Range Check** tab, enable the test, and enter the range of values shown below. Set the tolerance to 0% of rows.



e. Select in the **Demo** Project the **Entities > Uk Orders 2004 > Attributes > Payment Method** attribute, right click, and then select **Edit DSD**.

f. Select the **Values Check** tab, enable the test and enter values to check as shown below.

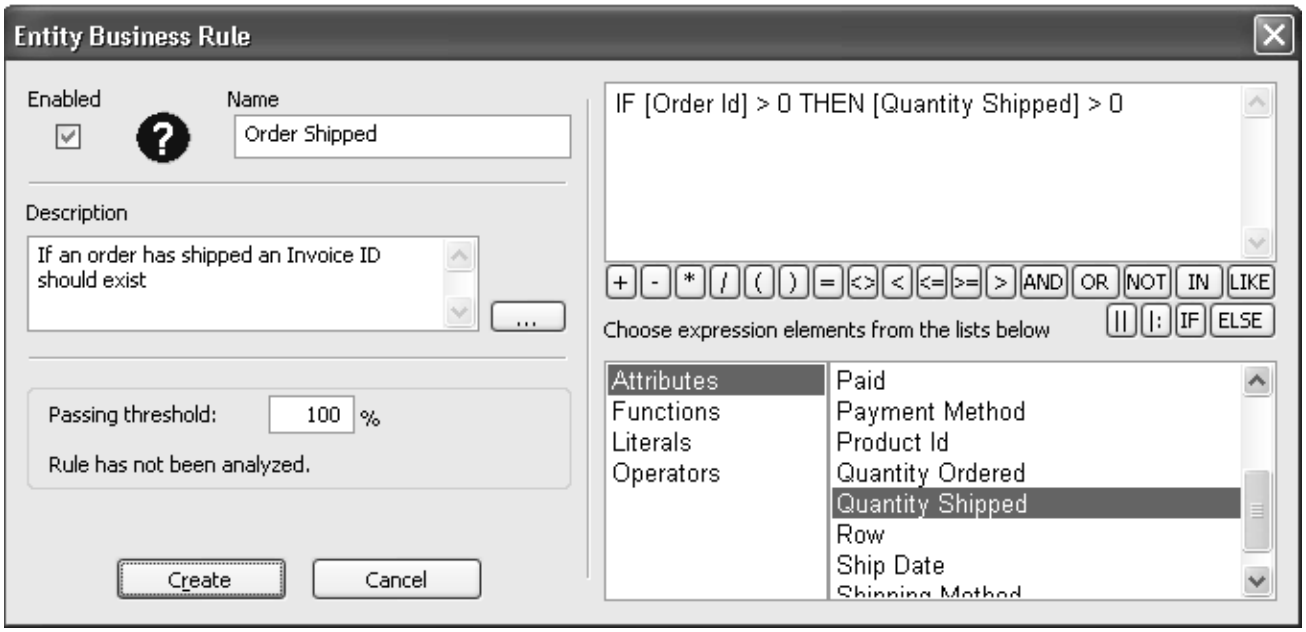


2. Re-analyze each Attribute
    a. Select in the **Demo** Project the **Entities > Uk Orders 2004 > Attributes > Order Id** attribute, right click, and then select **Re-Analyze Attribute DSDs**.
    b. Click **Run Now** in the Schedule Job popup window.
    c. Repeat these steps for the **Payment Method** attribute.
3. To examine **Compliance %**, expand the **Order ID** and **Payment Method** nodes, and then click the **Compliance %** information. You can drill down to the different DSD Tests results.

Apply Business Rules

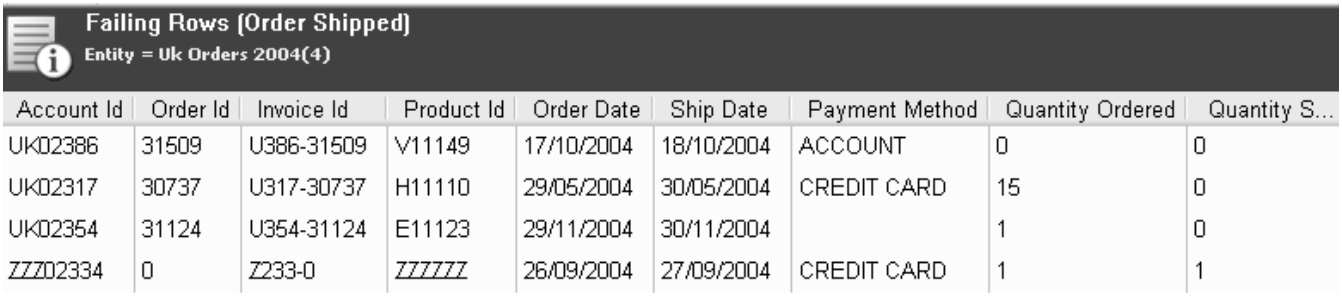
We now want to add a business rule to check the following business rule: "*If something was shipped then an order should exis*t".

1. Select in the **Demo** Project the **Entities > Uk Orders 2004 > Metadata > Business Rules**. Right-Click and select **Add Business Rule**.
2. Enter the business rule parameters as shown below. The code of the rule is :
   `IF [Order Id]>0 THEN [Quantity Shipped]>0`



3. Click on **Create**.

4. After creating the rule, you are prompted to check the rule. Click **OK** to check the run and **Run Now** to run the job immediately.
5. Double click the **Business Rules** node to list the business rules, and then double-click on the *Order Shipped* business rule to drill down to the failing rows. These show
    - 3 empty shipments, where *Quantity Shipped=0* and *Orders Ids > 0*
    - One shipment with no order, where *Quantity Shipped>0* and *Order Id = 0*

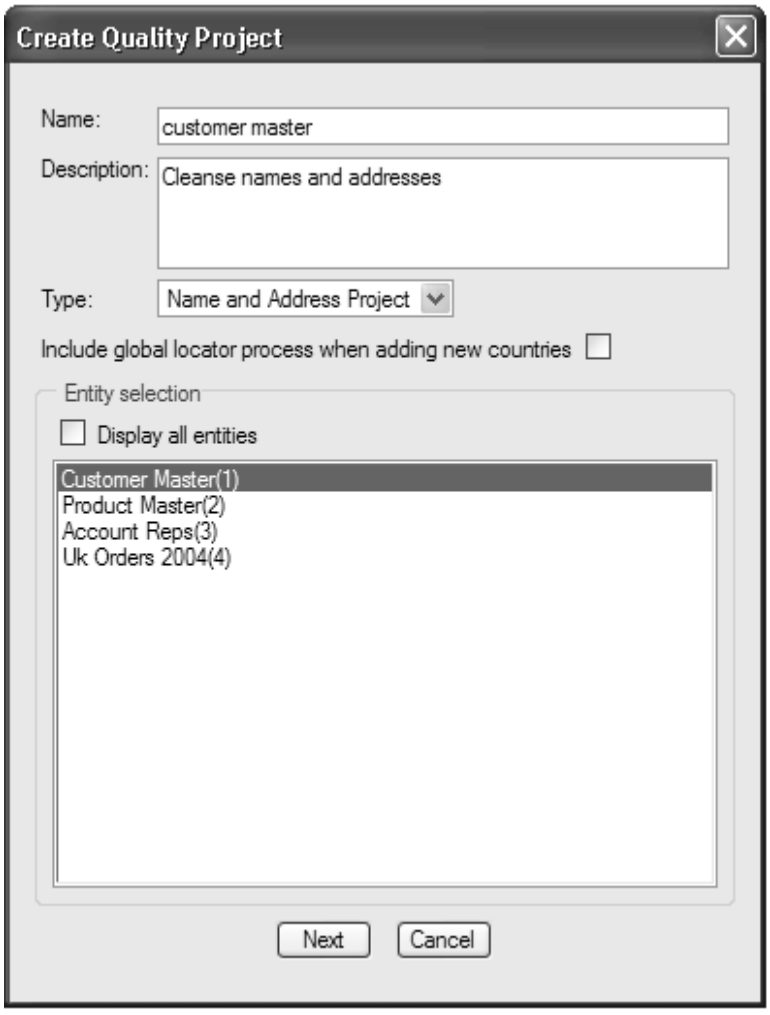


Failing Rows (Order Shipped)
Entity = Uk Orders 2004(4)

Account Id	Order Id	Invoice Id	Product Id	Order Date	Ship Date	Payment Method	Quantity Ordered	Quantity S...
UK02386	31509	U386-31509	V11149	17/10/2004	18/10/2004	ACCOUNT	0	0
UK02317	30737	U317-30737	H11110	29/05/2004	30/05/2004	CREDIT CARD	15	0
UK02354	31124	U354-31124	E11123	29/11/2004	30/11/2004		1	0
ZZZ02334	0	Z233-0	ZZZZZZ	26/09/2004	27/09/2004	CREDIT CARD	1	1

Design a Name and Address Cleansing Project

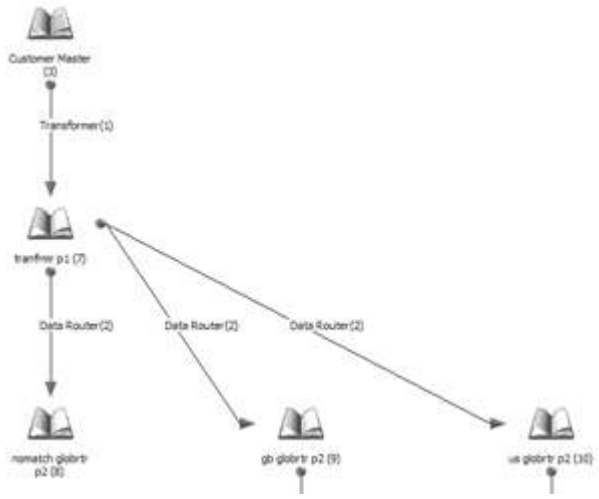
1. A data cleansing task is created in the form of a Quality project.
   Create a Quality Project as follows.
   a. Select **Quality** in the **Explorer** then right click and select **Create project…** in the popup menu.
   b. Enter the project name (**customer master**) and description, and then select **Name and Address Project**.
   c. Select the *Customer Master* entity, and then click **Next**.



   d. Add *United States (us)* and *United Kingdom (gb)* for the **Countries** and then click **OK**.

e. Click **Run Now** in the Schedule Job window.
Wait until the project is created, you can follow the project creation progress in the **Background Tasks** panel.

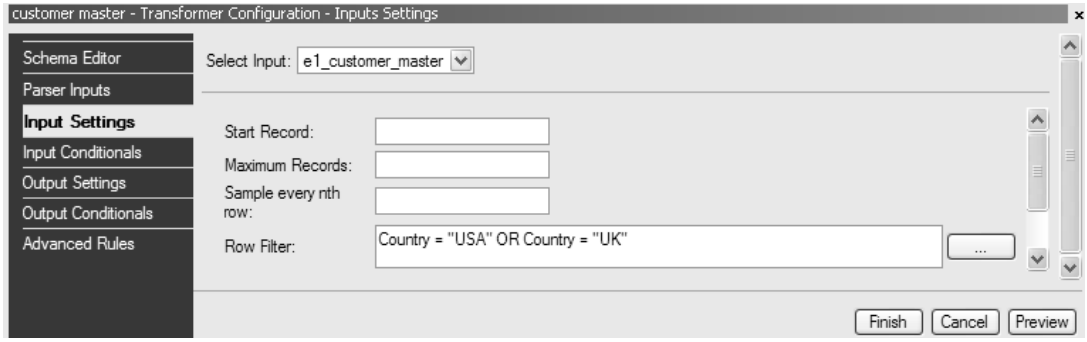
f. Double-Click on the *customer master* project under the **Quality** node. The project diagrams opens.



In this diagram, the arrows correspond to processes of the data cleansing project, and the books icons to the intermediate entities.

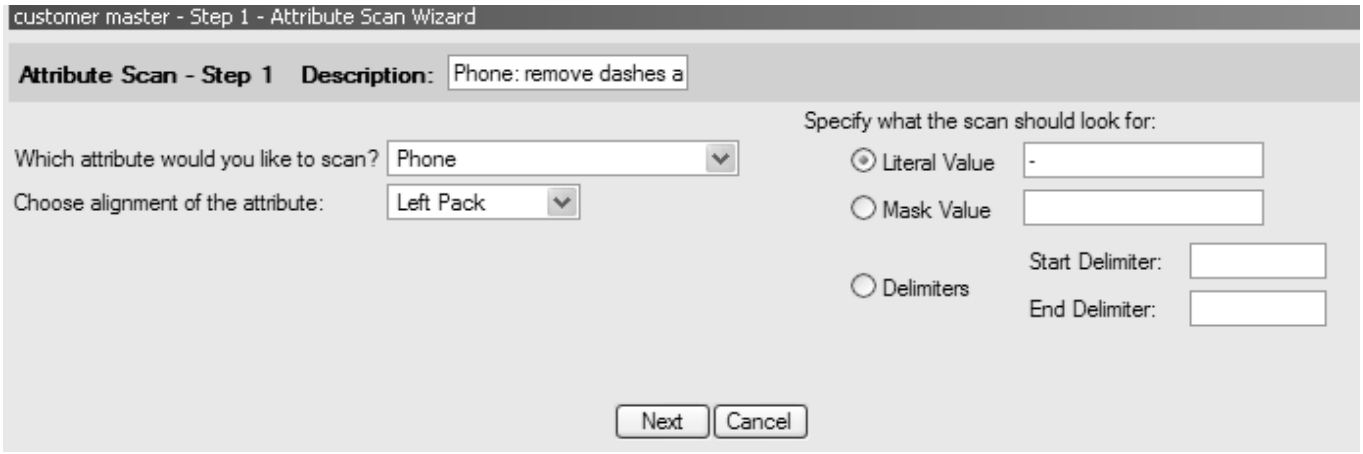
In this tutorial, we will review the processes of the data quality project, change and execute them step by step.

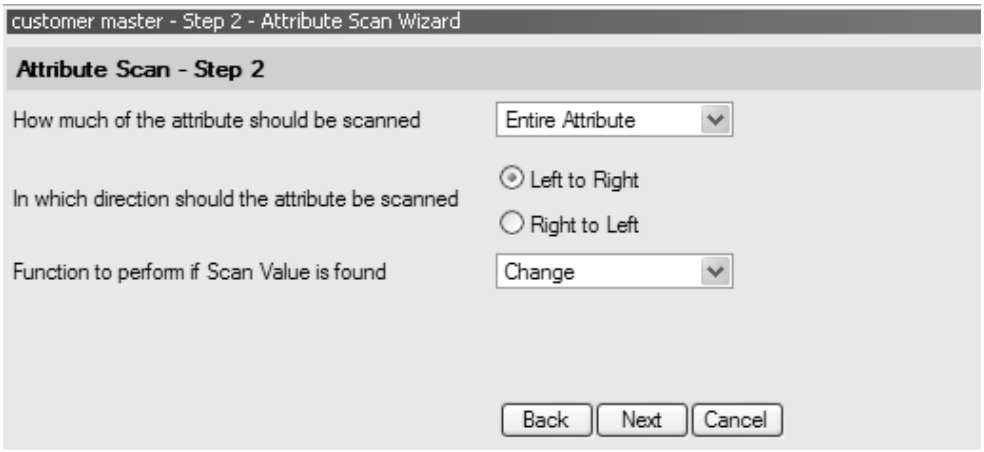
2. A **Transformer** process filters and performs basic transformations on input data. We use in our project a transformer to filter UK and US data and remove dashes and spaces from the phone numbers.
    a. Double click on the **Transformer** arrow to configure the **Transformer** process as follows:
    b. To filter US and UK data, define the following row filter in **Input Settings**:

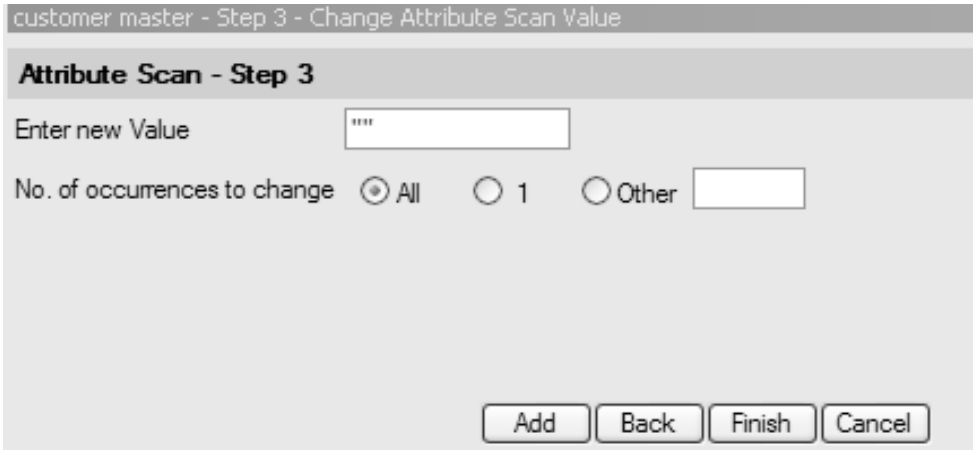


    c. To remove all dashes and spaces in the phone field, select the **Output Conditionals** option, then in the empty table, right click and select **Insert > New > Attribute Scan** in the popup menu.
        - **Description of scan**: Phone: remove dashes and spaces
        - **Which Attribute would you like to scan**: Phone
        - **Choose alignment of the attribute**: Left Pack - this option removes all spaces in the value.
        - **Specify what the scan should look for**: Literal Value.
        - **Literal Value**: - (dash symbol)
        - Change all instances of the value to :"" (two double quotes)
        - **No. of occurrences to change**: All

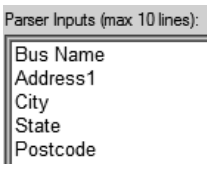
The wizard steps are given below:







d. In the same transformer, we will now define the relevant input fields that will be used by the country-specific address parsers. These fields can be overridden later in the country-specific transformers.
Under **Parser Inputs**, define your **Parser Inputs** as below



Note: For this tutorial we only take into account Business Names and disregard any personal name.

e. If you click the **Postcard…** button, you have a preview of the name and addresses fields that will be used for standardization.

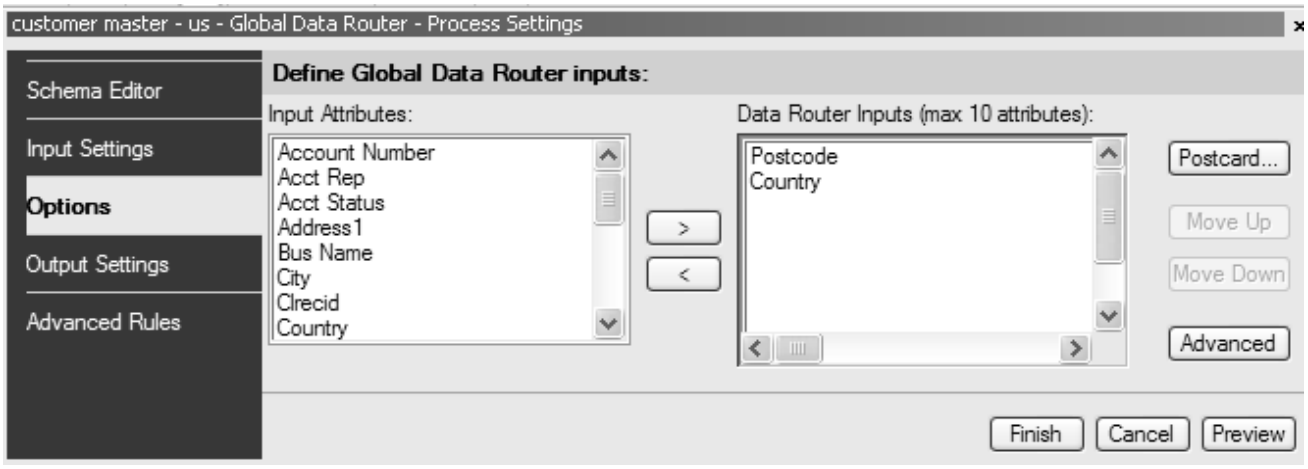
f. Click **Finish** to save the transformer parameters.
We need to configure the **Data Router** step prior to executing the Transformer step.

3. A **Global Data Router** separates the input records into separate entities depending on the country. As name and address processing is country-dependant, the router appears early in a data quality project.

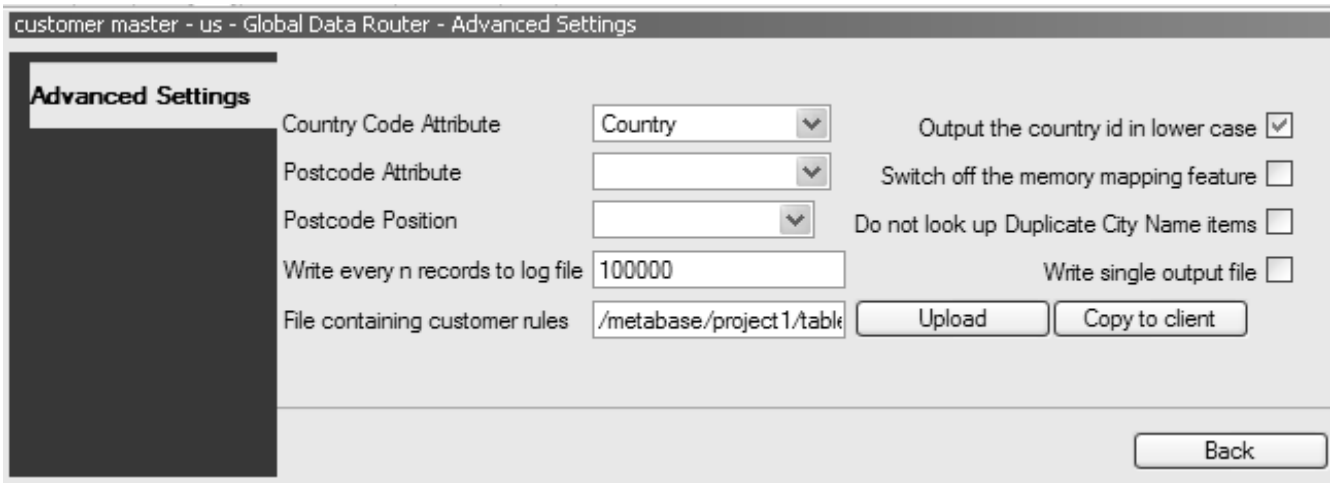
   a. Go back to the Quality project Diagram.

   b. Double click on the **Data Router** process.

   c. Under **Options**, select **Postcode** and **Country** as **Data Router Inputs.**



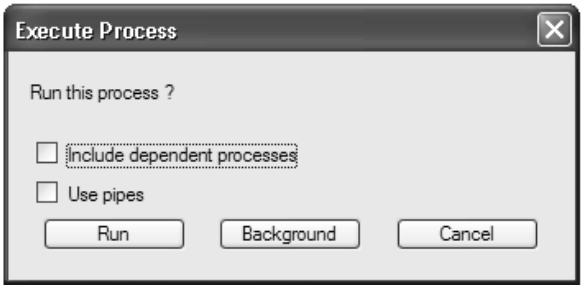
   d. Click the **Advanced** button and next to **Country Code Attribute**, select Country from the list.



   e. Click **Back** then **Finish**.

4. We can now execute the **Transformer** step.
Right-click on the transformer arrow in the diagram, and then select **Run**.

   a. In the Execute Process window, click on **Run**. Leave 'Include dependent processes' and 'Use pipes' unchecked.



b. Once the process has finished, right-click on the transformer arrow, then select **View…>Stats File**

The statistic file report appears as below.

```
RECORD INPUT
Count   Statistic               Qualifier       File
786     Records read    e1_customer_master
        C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E1/e1.dat
554     Records selected        e1_customer_master
        C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E1/e1.dat
0       Records bypassed        e1_customer_master
        C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E1/e1.dat
554     Records processed       e1_customer_master
        C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E1/e1.dat

RECORD OUTPUT
Count   Statistic               Qualifier       File
554     Records processed       OUTPUT
        C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E5/e5.dat
554     Records selected        OUTPUT
        C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E5/e5.dat
0       Records bypassed
        OUTPUT  C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E5/e5.dat
554     Records written OUTPUT
        C:/Oracle/product/11.1.1/odidq_1/oracledq/metabase_data/metabase/oracledq/E5/e5.dat

FIELD SCANNING STATISTICS
Count   EntryID Description     Format  Funct   Scan Field
289     1       Phone: remove dashes and spaces L       CH      PHONE
```

You can read the following statistics.

- In the *RECORD INPUT* section, *786 Records read* original input records. This results also in 554 records in the RECORD OUTPUT section.
- In the RECORD INPUT section, *554 Records selected* after the filter.
- In the FIELD SCANNING STATISTICS, *289 PHONE* fields scanned and transformed.

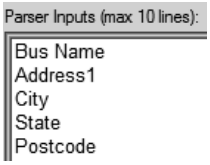
5. Right-click on the **Data Router** arrow in the diagram, and then select **Run**.
   a. In the Execute Process window, click on **Run**. Leave 'Include dependent processes' and 'Use pipes' unchecked.
   b. Once the process has finished, right-click on the **Data Router** arrow, then select **View…>Stats File**.
   These stats show 300 records for the USA and 254 for the UK (GB). Thanks to the filter in the transformer, there is no record with a NOMATCH qualifier.

```
RECORD INPUT
Count   Statistic              Qualifier
554     Records read           tranfrmr_p1(2)
554     Records selected       tranfrmr_p1(2)
0       Records bypassed       tranfrmr_p1(2)
554     Records processed      tranfrmr_p1(2)

RECORD OUTPUT
Count   Statistic              Qualifier
0       Records processed      NOMATCH C:/Docu
0       Records selected       NOMATCH C:/Docu
0       Records bypassed       NOMATCH C:/Docu
0       Records written        NOMATCH C:/Docu

300     Records processed      US      C:/Docu
300     Records selected       US      C:/Docu
0       Records bypassed       US      C:/Docu
300     Records written        US      C:/Docu

254     Records processed      GB      C:/Docu
254     Records selected       GB      C:/Docu
0       Records bypassed       GB      C:/Docu
254     Records written        GB      C:/Docu
```

6. Now that the data flow is split per country, we can perform country specific transformations using **Country-Specific Transformers**. In this specific transformer, we will configure in the **Parsers Inputs** which fields in the input records need to be examined when standardizing name and addresses. In our case, these fields are Bus Name (Business Name), Address1, City, State and Postcode (Zip Code).

Note: The Country Specific Transformer can be used to perform other type of transformations, such as the attribute scans we have used in the first transformer step.

   a. Double click on the **us Transformer** to edit it.
   b. Under **Parser Inputs**, check that the **Parser Inputs** are defined as below. These are inherited from the values specified in the first Transformer.



   c. If you click the **Postcard…** button, you have a preview of the name and addresses fields that will be used for standardization. Note that these are only US addresses.
   d. Click **Finish** to save the transformer parameters.
   e. Execute the **us Transformer** and examine the stats file. It should show that all 300 input records end up in the output records.

Note: For this tutorial, we will only focus on the US data, and delete all subsequent process steps involving UK data.

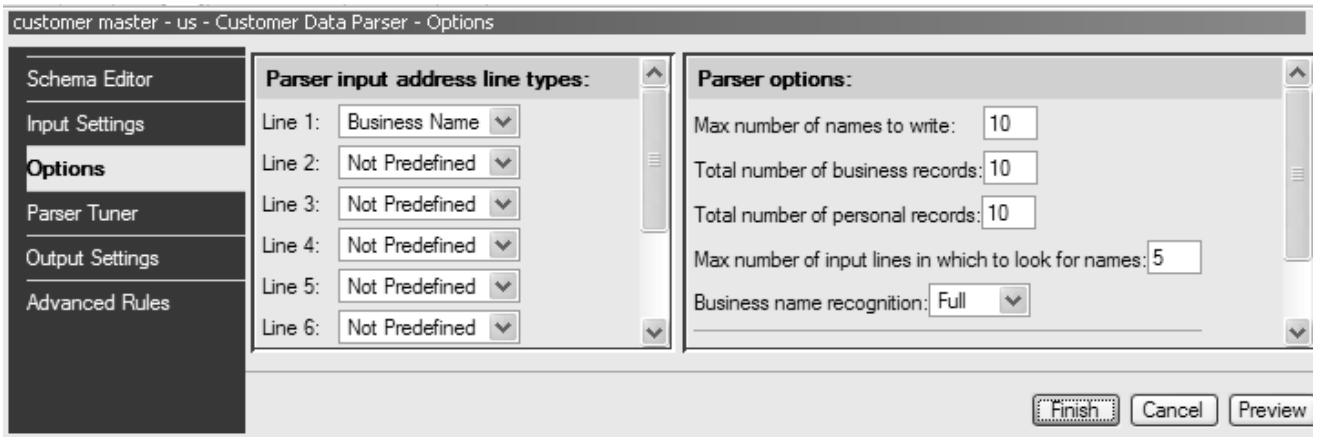
   f. Right-click on the **gb Transformer** step, then select **Delete Process… > This process and dependents**. Click OK to confirm and wait until all processes after the **gb globrtr pXX** entity are deleted.

7. We have defined the fields useful for recognizing the name and addresses. These fields will be analyzed by a **Customer Data Parser** that will identify and parse name and address data. This parser uses country-specific rules for analyzing the addresses. Its output is composed of the original data plus recoded or standardized data.
   We will customize the data parser by indicating that the first line returned by the previous

transformer step is a business line, and indicate that we only want to have one business name per record.

a. Double-click on the **us Customer Data Parser** process to edit it.

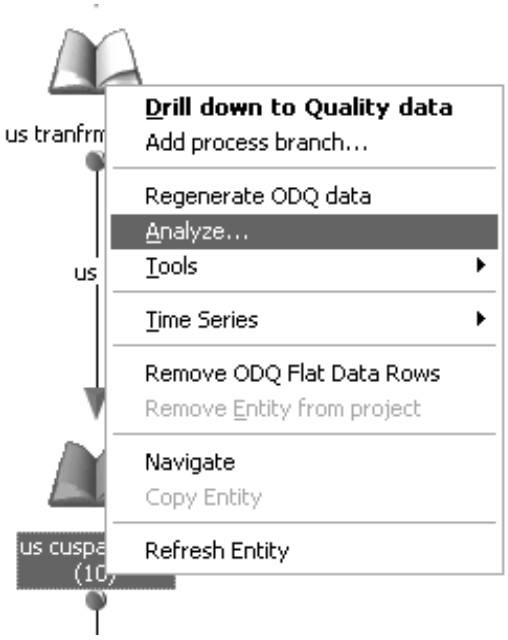
b. Select the **Options**, and then select *Business Name* for **Line 1**. Leave all other lines as *Not Predefined*.



c. Click **Finish** to apply your changes.

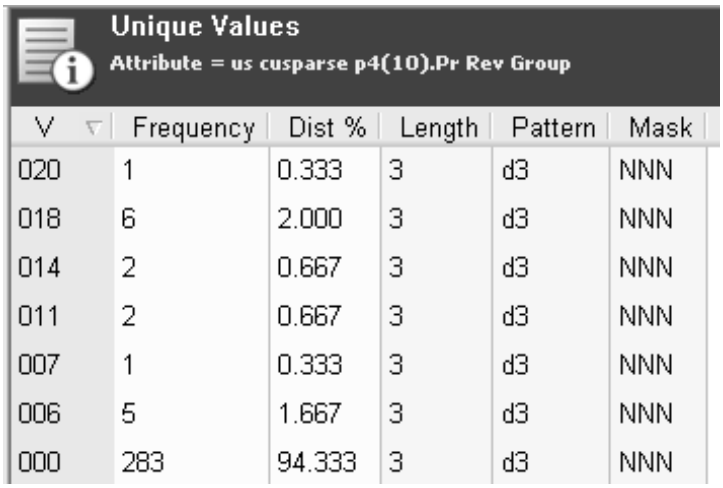
d. Execute the **us Customer Data Parser** process.

e. Right-click on the **us cusparse pXX** entity displayed under the **us Customer Data Parser**, then select **Analyze** in the popup menu.



f. In the window that appears, click **OK** to start the output entity analysis. Click on **Run Now** to execute the process.

g. In the Explorer (left panel), expand the **Quality > customer master > Entities > us cusparse pXX > Attributes > PR_REV_GROUP** nodes and double click on the **Unique Values** node.



Unique Values

Attribute = us cusparse p4(10).Pr Rev Group

V	Frequency	Dist %	Length	Pattern	Mask
020	1	0.333	3	d3	NNN
018	6	2.000	3	d3	NNN
014	2	0.667	3	d3	NNN
011	2	0.667	3	d3	NNN
007	1	0.333	3	d3	NNN
006	5	1.667	3	d3	NNN
000	283	94.333	3	d3	NNN

This value distribution shows the occurrence of the different data parser review group codes. For example, the 6 records with **Value=18** are those for which the city name is present but not recognized due to typos. You can drill-down and review the invalid values. See the on-line documentation for more information on the review codes and review group codes.

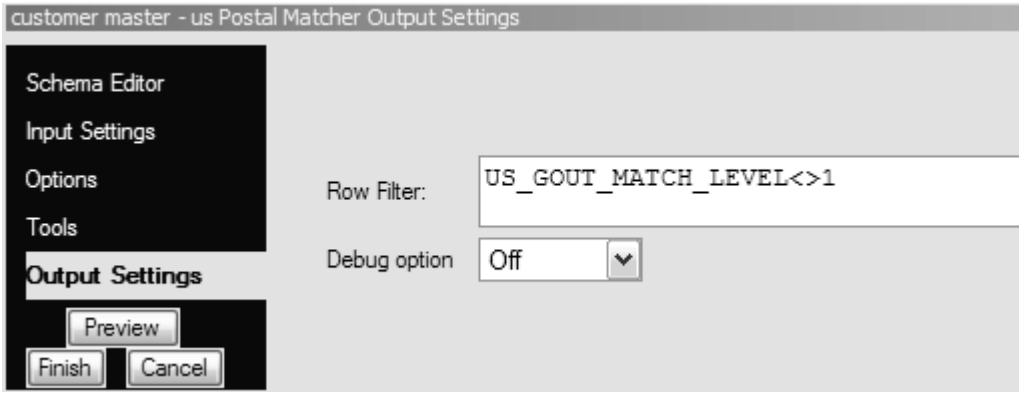
h. You can also examine the Stats file for the **us Customer Data Parser** process (right-click then View… > Stats File) to have a detailed report on the parsing.

8. The **Sort for Postal Matcher** sorts the data in geographic order to improve the performances of the next step: the postal matcher.
   a. Execute the **us Sort for Postal Matcher** process.
9. The **Postal Matcher** enriches data by matching data with postal directory information.
   a. Execute the **us Postal Matcher** process.
   b. Right-click on the **us pmatch pXX** entity displayed under the **us Customer Data Parser**, then select **Analyze** in the popup menu.
   c. In the window that appears, click **OK** to start the output entity analysis. Click on **Run Now** to execute the process.
   d. In the Explorer (left panel), expand the **Quality > customer master > Entities > us pmatch pXX > Attributes > US_GOUT_MATCH_LEVEL** nodes and double-click on the **Unique Values** node. These values correspond to how accurately the record matched with the postal directory data. Drill down to the rows for each value and examine them.
      - 0: exact match
      - 1: No city found to match.
      - 2: Street name failure
      - 3: House number range failure
      - 4: Street component failure
      - 5: Multiple possible matches to directory.

Important Note: Most records end up in a “1: No city found to match.” state. This is due to the fact that the sample postal directory used in this tutorial only contains postal information about New York City. Other cities are not recognized and the records cannot be enriched by the Postal Matcher. For a better readability of the results, we will filter the output of this process for the rest of the tutorial and ignore records outside New York.

e. Double-click the **us Postal Matcher** process to edit it.

f. Select the **Output Settings**, and then add a **Row Filter** as shown below.



g. Click **Finish** to save these settings, then execute the **us Postal Matcher** process again. The new *Record Output* section in the statistics (right-click **View … > Stats File**) for this process should now show *15 Records written.*

10. The **Window Key Generator** generates a composite key used in relationship linking. This key is constructed from elements contained in the input data. Records with similar Window Key are likely to be matching records. This key generation can be customized if necessary.

    a. Execute the **us Window Key Generator** process.

    b. Right-click on the **us winkey pXX** entity displayed under the **us Window Key Generator**, then select **Analyze** in the popup menu.

    c. In the window that appears, click **OK** to start the entity analysis. Click on **Run Now** to execute the process.

    d. In the Explorer (left panel), expand the **Quality > customer master > Entities > us winkey pXX > Attributes > WINDOW_KEY_01** nodes and double-click on the **Unique Values** node.

    e. Drill down to the unique values, then to the rows with matching WINDOW_KEY_01 values. These rows are likely to match (similar zip codes, business names, street address, person names, etc).

11. The **Sort for Linking** process sorts data for optimizing the relationship linker process.

    a. Execute the **us Sort for Linking** process.

12. The Relationship Linker process identifies records with a matching relationship or duplicate records. The output is categorized as success, fail, or suspicious based on the records similarity. A score is assigned to the records.

    a. Execute the **us Relationship Linker** process.

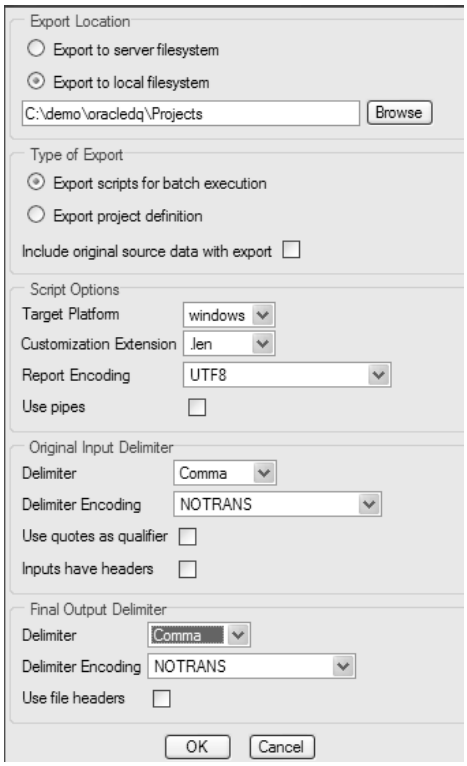
    b. Right-click on the **us rellink pXX** entity displayed under the **us Relationship Linker**, then select **Analyze** in the popup menu.

    c. In the window that appears, click **OK** to start the entity analysis. Click on **Run Now** to execute the process.

    d. In the Explorer (left panel), expand the **Quality > customer master > Entities > us rellink pXX > Attributes > LEV1_MATCHED** nodes and double-click on the **Unique Values** node.

    e. These unique values refer to the unique businesses detected by the relationship linker. If you click on one unique value, you see all rows representing the same business.

13. The **Commonizer** copies data across records linked by the relationship linker. It also selects a best of surviving record.
    a. Execute the **us Commonizer** process.
14. The **Transformer Address Reconstruction** reconstructs data for each output of the commonizer.
    Configure **us Transformer Address Reconstruction** as follows:
    a. Double-click on the **us Transformer Address Reconstruction** to edit it.
    b. Under **Schema Editor**, add the following to the Output Attributes list, by dragging them from the Attributes under **us common pXX** in the left of the panel at the end of the list of output attributes at the right of the panel.
        - **LEV2_SURVIVOR_FLAG**
    c. Click **Finish** to close the **us Transformer Address Reconstruction**
    d. Right-click on the **us Transformer Address Reconstruction** then select **Apply Schema Changes** in the popup menu.
    e. Click **OK** to apply the changes.
    f. Execute the **us Transformer Address Reconstruction** process.
    g. Double click on the output entity **us adtranfrmr pXX**, and examine the rows.
    h. Review the output records.
        - Records flagged *Us Gout Match Level = 0* are those that have been enriched with the postal data and those with *Us Gout Match Level = 2* are those that have not been correctly recognized and enriched.
        - Records flagged with *Lev2 Survivor Flag = 1* are the survivor records of the de-duplication process (containing the most comprehensive Business information).
        - The fields *Newaddr1, Newaddr2*, etc contain new address lines with comprehensive and cleansed addresses.
15. Export the project as a Batch Script. This process makes this project available for Oracle Data Integrator.
    a. From the **Explorer** or **Projects** panel, right-click on the **customer master** project and select **Export**.
    b. In the **Export Project Options** window, select **Export to local filesystem** and in the **Browse for Folder** window, select the **ODQ_SAMPLE_FILES**`\Projects` folder (create it if necessary) to export the project.
    c. Select your **Target Platform** (windows in this example), set the **Delimiter** to **Comma** for the **Original Input Delimiter** and to **Comma** for the **Final Output Delimiter**.



d. Click **OK**. A message indicates that the files are being copied.
This creates a folder called *projectN* (where *N* is the project identifier in Oracle Data Quality) and a *batch* sub-folder in the specified *Projects* output folder.
This batch sub-folder contains the following folders among others:

- **data**: This folder contains input and output data as well as temporary data files. As you specified No data for the export, this folder is empty for now.
- **ddl**: This folder contains the entities metadata files (.DDX and .XML). These files described the data files. These files are prefixed with `eNN_`, where NN is the Entity ID. Customized reverse-engineering is used in Oracle Data Integrator to retrieve these entities file format in the form of datastores.
- **scripts**: This folder contains the batch script `runprojectN.cmd`

e. First, we need to indicate to the data quality engine the location of the source and target files. The source file is referenced in the first transformer process settings file, and the target file is referenced in the last process settings.
Edit these files in a Text Editor (Notepad is not recommended).

- In the **settings** directory, open the file named `eN_ustranfrmr_pXX.stx` (where *N* is the internal entity reference of the entity corresponding to the *transfrmr pN* entity) and change the following options in the XML structured file:

  /CATEGORY/INPUT/PARAMETER/INPUT_SETTINGS/ARGUMENTS/ENTRY/DATA_FILE_NAME =
  ODQ_SAMPLE_FILES`\Data\customer_master.csv`

  To find the internal entity reference, click on Entities in the Explorer window.
  Then expand the *transfrmr pN* entity and double click on **Metadata**.
  Look for **Ref** and its corresponding to find the internal entity reference.

- Also in the **settings** directory, open the file whose name starts with e*N*_ (where *N* is the largest value in the directory) and ends with "*_delim.stx*" and change the following options in the XML structured file:

  /CATEGORY/OUTPUT/PARAMETER/OUTPUT_SETTINGS/ARGUMENTS/DATA_FILE_NAME =

  ODQ_SAMPLE_FILES\Data\cleansed_customer_master.csv

f. In the **scripts** folder, open `runprojectN.cmd` and comment out the following lines (add `::` in front of the commands):

```
:: convert output to csv
call "%TS_BIN%\tranfrmr" "%TS_SETTINGS%\e6_delim.stx"
call "%TS_BIN%\tranfrmr" "%TS_SETTINGS%\e8_delim.stx"
```

g. Your cleansing job can now be invoked by running the `<ODI_Home>\demo\oracledq\projects\oracledq\projects\oracledq\projectN\scripts\runProjectN.cmd`

Run the Quality Project in ODI

1. Open Oracle Data Integrator and connect to your repository. If you are starting with Oracle Data Integrator, use the demo environment.
2. In the Designer Navigator, select the Project view, click on the new project button to create a new project named *demo_quality*
3. Under this new project, open the **First Folder**, right-click on the **Packages** node and select **New Package**.
4. Enter *Quality Call* in the **Package Name** field, and then select the **Diagram** tab.
5. In the **Utilities** tools group in the toolbar, select the **OdiDataQuality** tool.
6. Click the diagram to add a step with this tool.
7. Set the following parameters for this tool:
   a. **Data Quality batch file**: Name of the `runprojectN.cmd` file in the `/scripts` sub-directory of your project export directory. For example: **ODQ_SAMPLE_FILES**`\Projects\project`*N*`\batch\scripts\runproject`*N*`.cmd`

      For example: `C:\demo\oracledq\Projects\project1\batch\scripts\runproject1.cmd`
2. Click the **Apply** button to save the package, then click on the **Execute** button to run it. The entire quality project runs.
3. You can go to **ODQ_SAMPLE_FILES**`\Data` to verify that a new cleansed_customer.csv has just been created by Oracle Data Quality.

Going Further with Oracle Data Quality for Data Integrator

Now that the Oracle Data Quality project runs, it is possible to use the input and output files in regular Oracle Data Integrator interfaces, in order to:

- Load the input file using datastores from various sources.
- Send the cleansed output data back into the source systems.