Record Matching: Tackling International Challenges with MatchUp

Balancing the process of selecting a matching criterion while achieving accurate results and performance.

Primary Issue: Deduplicating records in a varied format database.

Secondary Issue: Determining a strategy which will accurately identify duplicates (tight enough) but at the same time doesn't group false duplicates (too loose) or consume valuable resources (processing time and CPU usage).

Let's take a look at a few records which have entered a system with different formats and layouts.

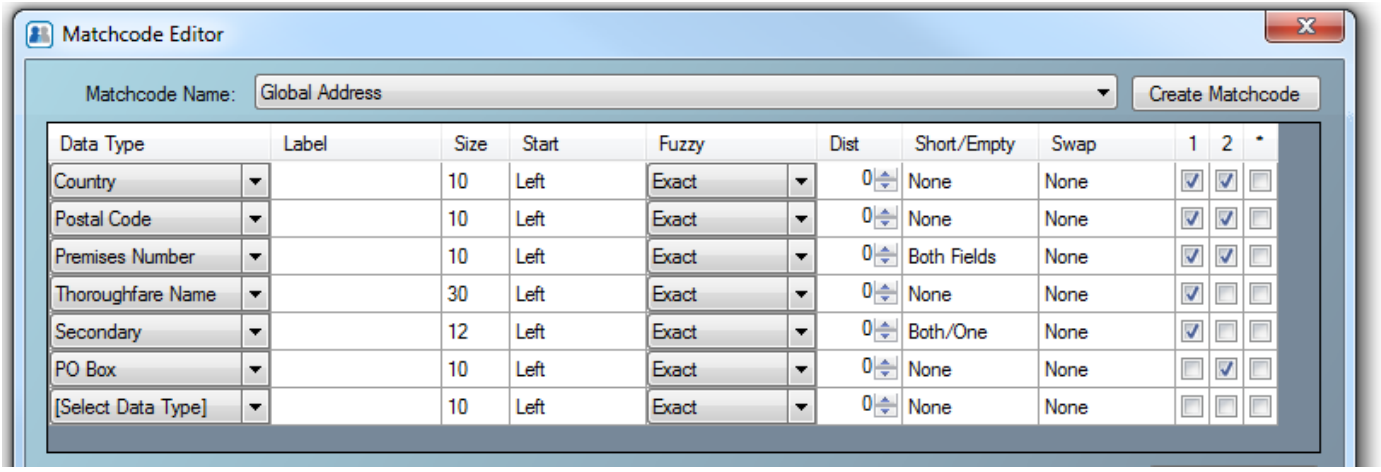
NAME	COMPANY	ADDRESS1	ADDRESS2	ADDRESS3	COUNTRY	ACCT	DATE
Herr Jürgen Smithe	Deutsche Bank GmbH	Suite 5	Berger Str. 130	60385 Frankfurt Am Main	DEU	400	2/13/2016
PHILIPP LAHM	FC Bayern München AG	Säbener Straße 51-57	81547	München	DEU		7/20/2008
PHIL LAHM	Bayern München	Säbener Str 51-57	81547 Münich		Germany	100	3/18/2012
Mr. J. Smithe	Deutsche Bank Ltd.	Berger Straße 130	60385	Frankfurt Am Main	DE	200	12/1/2005
Jürgen Smithe	Deutsche Bank	BergerStr 130	60385 Frankfurt		Germany	700	
Fräulein ERNA Keller		BÜRGERSTR. 2	DÜSSELDORF	40219	DEU	60	4/18/2014
HELEN LAHM	Bayern München	Säbener Str 51-57	81547 Münich		Germany	300	9/22/2014
HELEN ROLF	Bayern München		Säbener Str 51-57	81547 Münich	Germany	880	5/6/2011

International record matching presents a new set of challenges to MatchUp. Unlike domestic US / CAN, where Area Hierarchy and Postal code data (City, State, and Zip for example) appear in separate predefined database fields, global address elements can appear in up to eight address columns. How do we ensure that we accurately identify and remove existing duplicate records?

First we'll focus on the primary issue: identifying and removing duplicate records.

1. Select a matching strategy.

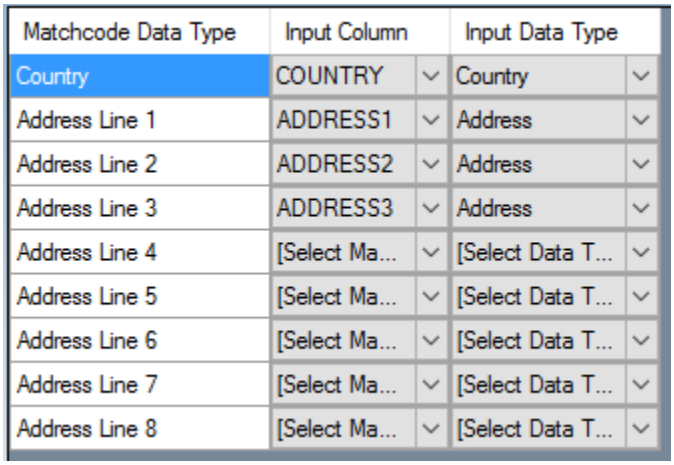
MatchUp is distributed with the Matchcode Editor – a 'Graphical User interface' which allows you to choose a pre-built ruleset called a matchcode, or create your own matchcode using a variety of input data types. Your business criteria will dictate the type of ruleset required to identify duplicate records in the database. We'll start with Global Address, a basic "Householding" (address) matching strategy which is distributed with Global MatchUp.



Selecting this matchcode will evaluate the components listed in the Data Type column for each record in your database.

The structure of our sample database is such that records with different layouts have found their way into the system. To account for these variations Matchup requires you to map in all columns which contribute to uniquely identifying records.

2. Field Mapping

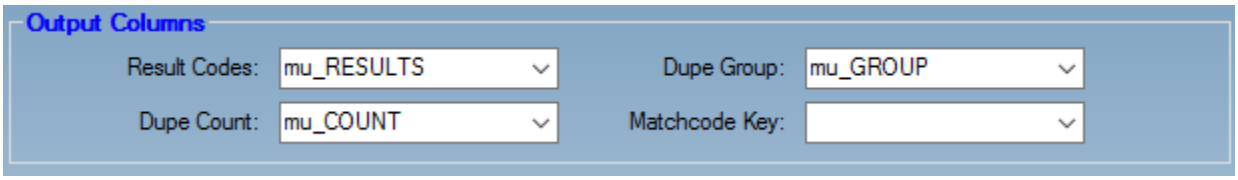


Map in your Country column and all columns that contain your address (including the Area Hierarchy) data.

For Global Matchcodes, MatchUp requires an input COUNTRY because it will tell MatchUp to recognize certain address tokens (like 'weg', 'straße', and 'Postfach' for example), and therefore identify address patterns for that country.
Also map in all columns that have address data. In this example we only have 3 address lines, so we map those and leave the remaining available address lines unmapped.

You also have to tell MatchUp the 'Input Data Type,' or the format of the data your source file is in. Here an Address Line contains an address, but for other Matchcode data types, like Names, the input data type may be any of various formats (like 'Smith, John,' which is an inverse format, or 'John Jr.,' which is a mixed first format). MatchUp needs this information in order to accurately build a representation for each record based on your matchcode, store these matchkeys internally, and then compare these keys to each other when we process.

3. Configure Options



If we're processing millions of records, how do we output meaningful results? How does MatchUp determine which records to output and which ones to discard as duplicates? Before processing, create output columns for:
-Result Codes – this is a status marking which tells you if a record is unique or an output record of a duplicate group, or a duplicate.
-Dupe Count – this tells you how many records were matched into the same group of a particular record.
-Dupe Group – this is an assigned unique identifier for each matched group (whether the group has many records or just one) for the process.

4. Process.

mu_RESULTS	mu_GROUP	mu_COUNT	NAME	ADDRESS1	ADDRESS2	ADDRESS3	ACCT	ACCT_TOTAL
MS01	1	1	Fräulein ERNA Keller	BÜRGERSTR. 2	DÜSSELDORF	40219	60	60
MS02,MS06	2	3	Herr Jürgen Smithe	Suite 5	Berger Str. 130	60385 Frankfurt /	400	1300
MS03,MS06	2	3	Mr. J. Smithe	Berger Straße 130	60385	Frankfurt Am Ma	200	
MS03,MS06	2	3	Jürgen Smithe	BergerStr 130	60385 Frankfurt		700	
MS02,MS06	3	4	HELEN LAHM	Säbener Str 51-57	81547 Münich		300	1280
MS03,MS06	3	4	PHIL LAHM	Säbener Str 51-57	81547 Münich		100	
MS03,MS06	3	4	HELEN ROLF		Säbener Str 51-57	81547 Münich	880	
MS03,MS06	3	4	PHILIPP LAHM	Säbener Straße 51-57	81547	München		

Now that MatchUp has linked our records into groups, it's time to evaluate those options using the result codes. Since 'MS01' represents a unique record, 'MSO2' represents the selected Output record from a group of duplicates, and 'MS03' represents a duplicate record, you can create a clean (deduped) output file by filtering MS01 and MS02 records. All possible returned Result Codes can be found here:

http://wiki.melissadata.com/index.php?title=Result_Code_Details#MatchUp_Object

Since an actual process may contain millions of records, you can use the Dupe Group property to link matching records. After updating your master database with the output results, found matches aren't always easily identified – you're not going to visually analyze thousands of rows, so using the Dupe Group to create or help maintain a group identifier makes it easy to locate or sort records that match.

Furthermore, the Count property can tell you how many matched records there actually are in that database for a particular group. This can also be useful in determining how clean (of duplicates) your master database actually is. Large dupe groups under the right criteria can mean that system data entry rules need to be revisited.

5. Further: Optimizing and Troubleshooting Your Matching Strategy

Now let's look at some of the issues you may see with the results. You may say, "wait, I can identify records that were returned as matches, but clearly, they are different contacts." Yes, in this case MatchUp placed different contacts in the same group - because we decided to use a "Householding" matchcode. Edit your matchcode by adding a Last Name component.

Data Type	Label	Size	1	2
Country		10	☑	☑
Last Name		10	☑	☑
Postal Code		10	☑	☑
Premises Number		10	☑	☐
Thoroughfare Name		30	☑	☐
Secondary	subpremise	12	☑	☐
Post Box		10	☐	☑
[Select Data Type]		10	☐	☐

Including the Last Name component will now give us an 'family level' matchcode strategy. And, thus, will require editing to our mappings…

Matchcode Data Type	Input Column	Input Data Type
Country	COUNTRY	Country
Last Name	NAME	Full Name
Address Line 1	ADDRESS1	Address
Address Line 2	ADDRESS2	Address
Address Line 3	ADDRESS3	Address
Address Line 4	[Select Ma...	[Select Data T...
Address Line 5	[Select Ma...	[Select Data T...
Address Line 6	[Select Ma...	[Select Data T...
Address Line 7	[Select Ma...	[Select Data T...
Address Line 8	[Select Ma...	[Select Data T...

Again, we see that for non-address components, there is a one-to-one mapping, but for our Address lines, we map in all columns that have address data.

MatchUp has internally parse out the necessary parts and disregarded discrepancies like Name Prefix, First Name, Middle Name, and Name Suffix – all of which were not specified in the matchcode. This allows us to match DIFFERENTLY FORMATTED names which have different tokens but are clearly duplicates.

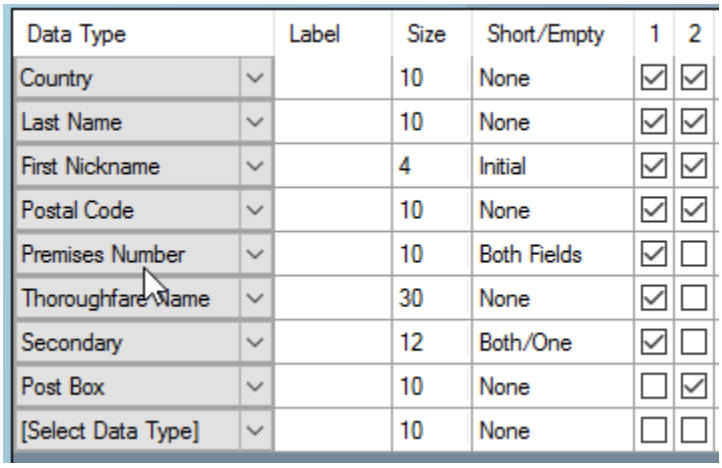
mu_RESULTS	mu_GROUP	mu_COUNT	NAME	ADDRESS1	ADDRESS2	ADDRESS3	COUNTRY
MS01	1	1	Fräulein ERNA Keller	BÜRGERSTR. 2	DÜSSELDORF	40219	DEU
MS02,MS06	2	3	HELEN LAHM	Säbener Str 51-57	81547 Münich		Germany
MS03,MS06	2	3	PHIL LAHM	Säbener Str 51-57	81547 Münich		Germany
MS03,MS06	2	3	PHILIPP LAHM	Säbener Straße 51-57	81547	München	DEU
MS01	3	1	HELEN ROLF		Säbener Str 51-5	81547 Münich	Germany
MS02,MS06	4	3	Herr Jürgen Smithe	Suite 5	Berger Str. 130	60385 Frankfurt Am Main	DEU
MS03,MS06	4	3	Mr. J. Smithe	Berger Straße 130	60385	Frankfurt Am Main	DE
MS03,MS06	4	3	Jürgen Smithe	BergerStr 130	60385 Frankfurt		Germany

Now we've placed different individuals in the same family (by last name) at the same address into different groups. We can take this one step further by changing the matchcode to include a First Name component.

Matchcode Data Type	Input Column	Input Data Type
Country	COUNTRY	Country
Last Name	NAME	Full Name
First Name	NAME	Full Name
Address Line 1	ADDRESS1	Address
Address Line 2	ADDRESS2	Address
Address Line 3	ADDRESS3	Address
Address Line 4	[Select Ma...	[Select Data T...
Address Line 5	[Select Ma...	[Select Data T...
Address Line 6	[Select Ma...	[Select Data T...
Address Line 7	[Select Ma...	[Select Data T...
Address Line 8	[Select Ma...	[Select Data T...

mu_RESULTS	mu_GROUP	mu_COUNT	NAME	COMPANY	ADDRESS1	ADDRESS2	ADDRESS3	COUNTRY
MS01	1	1	Fräulein ERNA Keller		BÜRGERSTR. 2	DÜSSELDORF	40219	DEU
MS01	2	1	HELEN LAHM	Bayern München	Säbener Str 51-57	81547 Münich		Germany
MS01	3	1	PHIL LAHM	Bayern München	Säbener Str 51-57	81547 Münich		Germany
MS01	4	1	PHILIPP LAHM	FC Bayern München AG	Säbener Straße 51-57	81547	München	DEU
MS01	5	1	HELEN ROLF	Bayern München		Säbener Str 51-57	81547 Münich	Germany
MS01	6	1	Mr. J. Smithe	Deutsche Bank Ltd.	Berger Straße 130	60385	Frankfurt Am Main	DE
MS02,MS06	7	2	Herr Jürgen Smithe	Deutsche Bank GmbH	Suite 5	Berger Str. 130	60385 Frankfurt Am Main	DEU
MS03,MS06	7	2	Jürgen Smithe	Deutsche Bank	BergerStr 130	60385 Frankfurt		Germany

We've accurately put contacts with obvious differences in First Name into another group, but discrepancies in the First Name are placing true duplicates into different groups. MatchUp has advanced matchcode data type settings to catch these, so we'll make a few final changes.



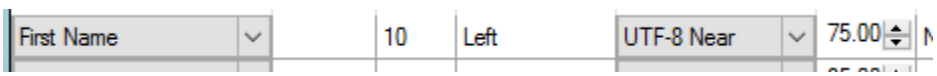
By replacing the First Name with the First Nickname data type and allowing a spelled out First Name to match an initial, we can catch names like 'D Smith' to 'Don Smith', and 'Aimee Reed' to 'Amy Reed'.

mu_RESULTS	mu_GROUP	mu_COUNT	NAME	COMPANY	ADDRESS1	ADDRESS2	ADDRESS3	COUNTRY
MS01	1	1	Fräulein ERNA Keller		BÜRGERSTR. 2	DÜSSELDORF	40219	DEU
MS02,MS06	2	2	PHIL LAHM	Bayern München	Säbener Str 51-57	81547 Münich		Germany
MS03,MS06	2	2	PHILIPP LAHM	FC Bayern München AG	Säbener Straße 51-57	81547	München	DEU
MS01	3	1	HELEN LAHM	Bayern München	Säbener Str 51-57	81547 Münich		Germany
MS01	4	1	HELEN ROLF	Bayern München		Säbener Str 51-57	81547 Münich	Germany
MS02,MS06	5	3	Herr Jürgen Smithe	Deutsche Bank GmbH	Suite 5	Berger Str. 130	60385 Frankfurt Am Main	DEU
MS03,MS06	5	3	Mr. J. Smithe	Deutsche Bank Ltd.	Berger Straße 130	60385	Frankfurt Am Main	DE
MS03,MS06	5	3	Jürgen Smithe	Deutsche Bank	BergerStr 130	60385 Frankfurt		Germany

Even further: What if we had typos or a need to apply different fuzzy logic to more accurately catch duplicates? Example:

Jurgen Doe and Jürgen Doe

By changing our matchcode to use a Fuzzy algorithm on the First Name component instead of an 'exact' setting, like this…

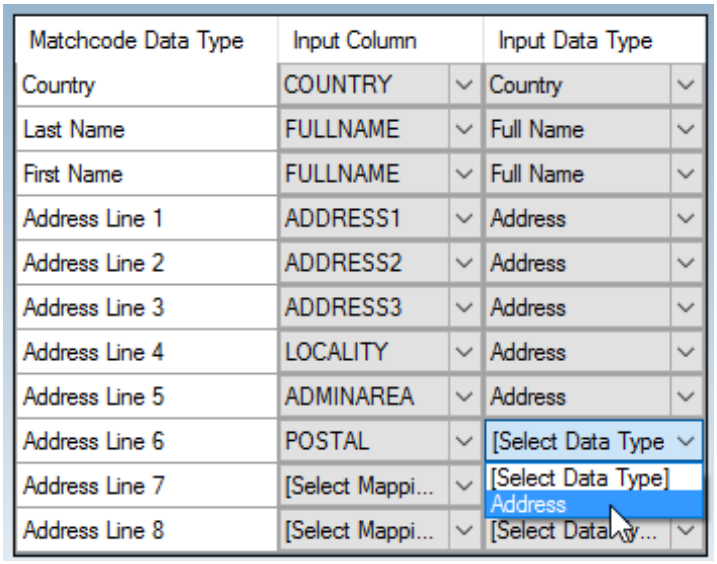


…you will now group these together (if the two strings are found to be 75% or more similar). Fuzzy algorithms can be used on many different data types.

CONSIDERATIONS: In some languages, it may be important to distinguish accented characters as distinct (for example, due to gender differentiation or completely different names). In these cases, knowing the source data encoding and using an 'Exact' setting may be proper configuration.

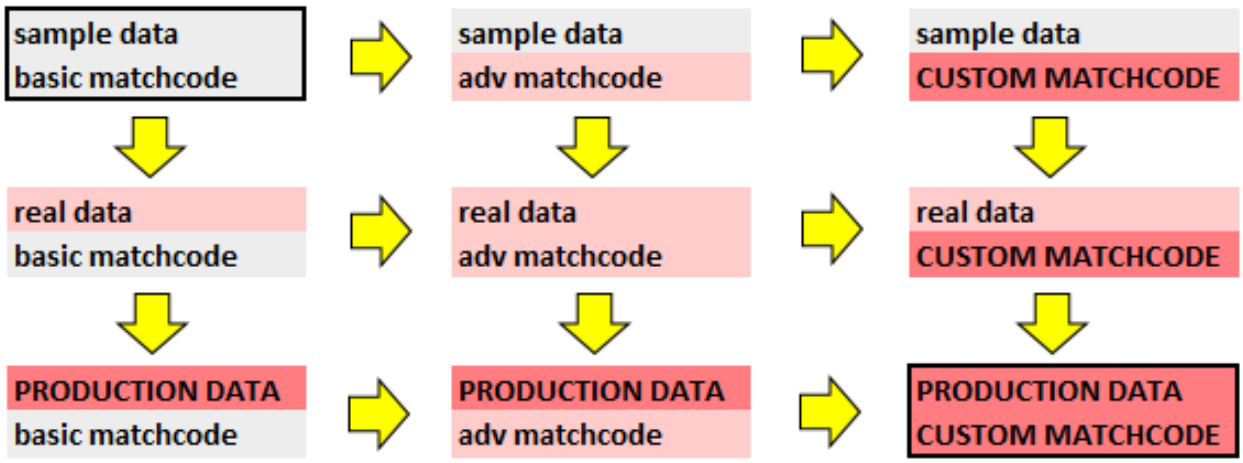
Also, keep in mind that when applying advanced matchcode settings (whether that may be fuzzy algorithms, multiple combinations of conditions, etc.), you are asking matchup to perform algorithmic computations for every key being compared. This can potentially return more duplicates, but can exponentially increase processing time for a particular job. Further Advanced Concepts discussions can be found here:
http://wiki.melissadata.com/index.php?title=MatchUp_Object

Returning to our mappings: what if I am adding matching by name, but have already parsed Area Hierarchy data? In this case, MatchUp still requires all your address data mapped into the full address lines, so map them in as such:



You'll notice that MatchUp will let you map in these distinct columns, but your only choice as input data type is 'Address.' In this example, notice we also mapped out Last Name and First Name components differently. This database didn't have parsed out names, so we simply map in the FULLNAME source column and tell MatchUp, that this column contains Fullnames – it will identify the Last and First names, and build the keys accordingly.

Conclusion:
Matching database records from a variety of countries requires an understanding of the different formats and postal standards. Fortunately, MatchUp and our underlying Global Address engine simply require an input country designator and all Address columns to return accurate record matching. But to achieve the best level of record matching beyond the address requires identifying not only the format of the desired components, but also their quality. Here, for example, adding levels of name matching had allowed us to break initial family groups into smaller individual level groups. Repeating the cycle of testing the matchcode, analyzing the results, and fine-tuning a strategy may, as shown, require you to implement other matching techniques at the matchcode component level to arrive at the degree of matching accuracy required by your production environment.
For new MatchUp users (as well as those who simply want to refine their match rules) we always recommend this multi-step strategy.



Taking this approach of 'small steps' over blindly 'throwing mud at the wall' from the outset with a very advanced matchcode will not only save development time, but gives you a better understanding of how an implemented matching strategy relates to the returned process results.

NOTES:

For MatchUp Object users, the concepts and steps here are the same, but you will be programmatically calling the respective methods and retrieving output properties after processing.

In addition to the benefits of Domestic and or Intl Record processing, the ETL solution can be easily configured to provide:
1.Golden Record (or Record Prioritization) based on a pecking order you provide.
This lets you determine which record to keep, and which to flag as duplicates.
2.Survivorship (or Record Rollup, or Data Gathering) in a variety of available methods
This allows you to consolidate data from grouped records into that single output record.
3.ResultCode driven Output Streams
This allows you to output and send processed records into different output streams based on our result codes, thus saving you having to write custom queries.