How to generate a Direct Match Key

Prepared by **Statistics Canada**

June 2018

In partnership with Canada

When using administrative data to maintain a Statistical Business Register (SBR), there is often a need to link the two data sources together. This is typically done through a record linkage strategy that includes an automated process for detecting possible links based on linking criteria. To increase the chances of find links, prior to the linkage attempts are made to groom the linkage variables and derived variables are generated. This document outlines how a Direct Match Key (or derived version) of business names can be generated, and how it is used in record linkage.

A typical DMK algorithm first:

Converts all characters to uppercase.

Then it removes the following items:

Words that can have different abbreviations and do not add value to the name
(LTD, LTEE, LIMITED, INC, INCORPORATED, O/A, C/O, CO, COMPANY, AND, OF, THE)
Accents from letters,
Punctuation and spaces,
Double consonants,
"Y" at the end of a word or if it appears after a consonant,
"S" at the end of a word,

The last step is to:

Truncate the name to the first 14 characters.

Appendix A of this document presents how the above algorithm could be programmed in SAS.

When using DMK values within a linkage, attempts are first made to link the business names then on the DMK business names. It is important to attempt to link on the original business names first in order to limit the chances of producing false links. The following table provides a couple of examples where names are easily recognized as being different, but produce the same DMK value.

Business Names	DMK Business Names
Catty's Food Emporium Ltd	CTFDMPRM
The Cat Food Emporium Inc	CTFDMPRM
Freddy and Rick's Engineers - Piers	FRDRCKNGNRPR
Fredrick's Engine Repair Limited	FRDRCKNGNRPR

The idea of the DMK is to produce a value that will overcome basic spelling errors and slight variations in spelling. However, it also introduces a high level of compression which removes meaning from the words and can create duplicated values from very different names.

As one can see from the SAS coding in Appendix A, the order in which the steps are executed can have an impact on the final outcome of some DMK values.

Appendix A
SAS Script of a DMK Procedure

```
/**ADD DMK TO THE FILE**/

/*setup an array of words that need to be removed*/
array words{7}$63. _temporary_ ('LTD','LTEE','LIMITEE','INC','INCORPORATED','O/A','C\O');

/*change the French letters to English letters*/
new_name=translate(upcase(REGISTERED_NAME),'CEEAUOUA','ÇÈÉÀÙÔÛÂ');

/*Remove the words that are in the array*/
Do I=1 to 7;
        tmp=words{I};
        new_name=tranwrd(new_name,trim(tmp),'');
end;

/*Split the name into words*/
array words2{10}$64 n1-n10;
Do I=1 to 10;
        call scan(new_name, I, position, length);
        if not position then leave;
        words2{I}=substrn(trim(new_name), position, length);
end;

/*Setup an array of consonants*/
array words1{21}$1. _temporary_ ('B','C','D','F','G','H','J','K','L','M','N','P','Q','R','S','T','V','W','X','Y','Z');
Do I=1 to 10;
        new_name=words2{I};

        /*remove punctuation marks and space*/
        new_name=compress(new_name,,"PS");

        /*if the word  is 'CO','COMPANY','AND','LIMITED'and 'OF', then remove*/
        if new_name eq 'CO'   or new_name eq 'COMPANY' or new_name eq 'THE' or
          new_name eq 'AND' or new_name eq 'LIMITED'     or new_name eq 'OF'
        then new_name='';

        /*remove 'Y' if it is at the end of a word, even after a vowel*/
        namelast=substrn(new_name,length(new_name),1);
        if namelast eq 'Y' then new_name=substrn(new_name, 1, length(new_name)-1);

        /*Extract the last letter of the word*/
        namelast=substrn(new_name,length(new_name),1);

        /*remove 'Y' if it is at the end of a word*/
        if namelast eq 'S' then new_name=substrn(new_name, 1, length(new_name)-1);
        /*remove double consonants, replace with single consonants*/
        Do j=1 to 21;
                tmp=words1{j};
                tmpdouble=trim(tmp)||trim(tmp);
                new_name=tranwrd(new_name,trim(tmpdouble),trim(tmp));
        end;
```

```
        /*Remove 'Y' if it is after a consonant*/
        Do j=1 to 21;
                tmp=words1{j};
                tmpy=trim(tmp)||'Y';
                new_name=tranwrd(new_name,trim(tmpy),trim(tmp));
        end;
        namefirst=substrn(trim(new_name),1,1);
        new_name=substrn(new_name,2,length(new_name)-1);

        /*remove vowel*/
        new_name=compress(new_name,"AEIOU");

        /*concatenate the first letter with the processed word*/
        final=trim(namefirst)||new_name;
        words2{I}=trim(final); end;

/*concatenate all words to form DMK*/
finalname=trim(words2(1));
do k=1 to 9;
        finalname=trim(finalname)||trim(words2{k+1});
end;

/*remove any leading and trailing blanks */
finalname=trim(left(finalname));

/*Keep only the first 14 characters */
DMK_EST=substrn(finalname,1, 14);
```