

ASCII TO DAIMOND CONVERTER

User Guide



Version No: 1.0 Release Date: 23rd May 2016



INDIA**Subex Limited**

Adarsh Tech Park
Outer Ring Road
Devarabisanahalli
Bangalore - 560,037, India
Tel: +91 80 6659 8700
Fax: +91 80 6696 3333
www.subexworld.com

CANADA**Subex Americas Inc.**

30 Fulton Way
Richmond Hill, Ontario
Canada L4B 1E6
Tel: +1 905 886-7818
Toll free: +1 877 899-4612
Fax: +1 905 886-9076

USA**Subex, Inc**

12101 Airport Way Suite 300
Broomfield, CO
Tel: +1 303 301 6200
Fax: +1 303 301 6201

UK**Subex (UK) Limited**

3rd Floor, Finsbury Tower
103-105 Bunhill Row
London, EC1Y 8LZ UK
Tel: +44 20 7826 5420
Fax: +44 20 7826 5437

SINGAPORE**Subex (Asia Pacific) Pte Limited**

Pte Limited
175A Bencoolen Street
#06-06/07 Burlington Square
Singapore 189650
Tel: +65 6338 1218
Fax: +65 6338 1216

DUBAI**Subex Limited Office**

1304, Al Shatha Tower,
Dubai Media City,
P.O. Box: 502520, Dubai,
United Arab Emirates
Tel: +971 4 3642591
Fax: +971 4 3697648

IMPORTANT NOTICE

This document constitutes confidential and proprietary information of Subex Limited. The contents of this document may be accessed and/or used solely by a licensee of Subex Limited software product(s) and solely in connection with the licensee's authorized use of such product(s), or as otherwise expressly permitted by Subex Limited in writing. All other uses are prohibited. This document may not in any event be disclosed to any third party without the prior written authorization of Subex Limited.

Subex™ is the trademark of Subex Limited. All other brands and product names identified in this publication are trademarks or registered trademarks of their respective companies or organizations.

Copyright© Subex Limited

All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems for any purpose without the express written permission of Subex Limited.

Purpose and Scope

This documents serves the Usage of ASCII to Diamond Converter.

Revision History

| Issue | Date | Author | Remarks |
|-------|------------|---------------------|-----------------|
| 1 | 22-05-2016 | Battula Sai Krishna | Initial Version |
| | | | |
| | | | |

Reference Documentation

| Date | Document Title |
|------|----------------|
| | |
| | |

Table of Contents

| | |
|--|-----------|
| 1. Overview of ASCII To Diamond Converter..... | 5 |
| 1.1. Executive Summary | 5 |
| 1.2. Audience..... | 5 |
| 2. Usage of ASCII To Diamond Converter..... | 5 |
| 2.1. Requirements..... | 5 |
| 2.2. Execution of ASCII To Diamond Converter | 6 |
| 2.3. Output..... | 9 |
| 3. Efficiency | 11 |
| 3.1. Normal Proof of Efficiency Improvement..... | 11 |
| 3.2. Algorithmic Proof of Efficiency Improvement | 12 |
| 3.2.1. Time Consumed by Manual Method | 12 |
| 3.2.2. Time Consumed by ASCII To Diamond Converter | 13 |
| 4. Limitation | 14 |
| 5. Sample Files..... | 14 |

1. Overview of ASCII to Diamond Converter

1.1. Executive Summary

The objective of this document is to provide guidelines that one needs to follow while using the ASCII to Diamond Converter to create an ASCII Decoder. This will help us in developing the ASCII Decoder in a faster way.

This document will be periodically updated based on the new versions released.

The document is divided into two key sections. The first section details on how to use the ASCII to Diamond Converter while the second section details on how it is efficient than normal process of creating ASCII Decoder.

1.2. Audience

This document is intended for the Subex Users who are ready to develop the ASCII decoder.

2. Usage of ASCII to Diamond Converter

This section describes how to use the ASCII to Diamond Converter.

2.1. Requirements

The following must be required to develop the ASCII Decoder without manual efforts

1. Java 1.7 must be installed in your machine

```
C:\Users\battula.krishna\Desktop>java -version
java version "1.7.0_80"
Java(TM) SE Runtime Environment (build 1.7.0_80-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.80-b11, mixed mode)
```

Java Version

2. AsciiToDiamondConverter_v1.0.jar file



AsciiToDiamondCo
nverter_v1.0.jar

3. Keep the parse fields in a file which is separated by a delimiter (, @ | # ! \$ ^ & * () \n(newline))
Please find the below figure where parse fields are taken where each field is separated by a new line (\n)

```
HLR_INDEX
IMSI
MSISDN
GPRS_SUBSCRIBED
APN
PDP_ADD
NAM
ODBIC
ODBOC
ODBPLMN1
ODBPLMN2
ODBPLMN3
ODBPLMN4
```

Parse Fields

4. {Optional requirement, if you want Mapping fingerprint to be created then this step needs to be followed} Keep the Mapping fingerprint fields in a file which is separated by a delimiter (, @ | # ! \$ ^ & * () \n(newline))

Please find the below figure where Mapping fingerprint fields are taken where each field is separated by a pipe(|) delimiter

```
fng_id|sfl_id|sts_id|record_address|record_length|record_number|record_type|hlr_filename|hlr_imsi|hlr_msisdn|hlr_services|hlr_ss|hlr_apn_id|hlr_calbinc|hlr_calboak|hlr_confcad|hlr_cusclip|hlr_cusclir|hlr_extrahlr|hlr_gprs|hlr_intcall|hlr_oick|hlr_csp|hlr_oick_csp|hlr_roaming|hlr_shortms|hlr_stype|hlr_voice|hlr_crbt|hlr_subscription_type|hlr_active_subscriber|hlr_dump_datetime|hlr_reserved1|hlr_reserved2|hlr_reserved3|hlr_reserved4|hlr_reserved5
```

Fingerprint fields

2.2. Execution of ASCII to Diamond Converter

1. Execute the AsciiToDiamondConverter_v1.0.jar in cmd Prompt using the below command

```
java -jar AsciiToDiamondConverter_v1.0.jar
```

```
C:\Users\battula.krishna\Desktop>
C:\Users\battula.krishna\Desktop>java -jar AsciiToDiamondConverter_v1.0.jar
```

2. Then it prompts for the parse fields filename with path.

For suppose my Parse fields file is located in

C:\Users\battula.krishna\Desktop\ASCII_TO_DAIMOND\Samples\ParseFields.txt

```
Enter the filename containing the list of parse input fields:
C:\Users\battula.krishna\Desktop\ASCII_TO_DAIMOND\Samples\ParseFields.txt
```

3. Then it will prompts for the delimiter for the parse fields file.

For suppose parse fields are separated by a new line (\n).

```
Enter the field delimiter for above file (new line delimiter = \n):
\n
```

- Then it will prompts for which type of delimited ASCII you wants to create.

For suppose my raw CDR's is separated by a comma (,) them my decoder will be a comma delimited ASCII

```
Please enter Which type of delimited ASCII (comma, space, tab, !, etc.,):
comma
```

In Diamond we will set delimiter for delimited ASCII as below,

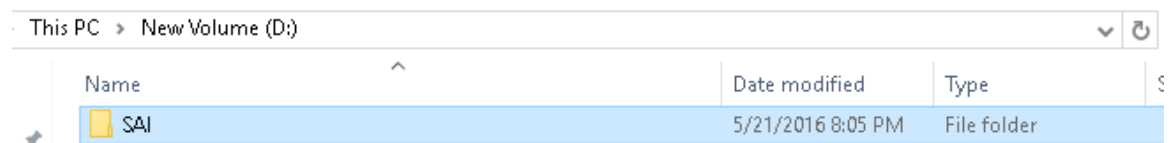
- Then it will prompt for a Directory Creation in D Drive (Enter the name of the directory to create in D drive where output decoder resides.)

For suppose I am creating SAI

```
Enter the name of a direccory to create in D:\ where output decoder Resides :
SAI
```

Note: Directory Name should not contain Space.

Then the directory SAI will be created in D Drive.



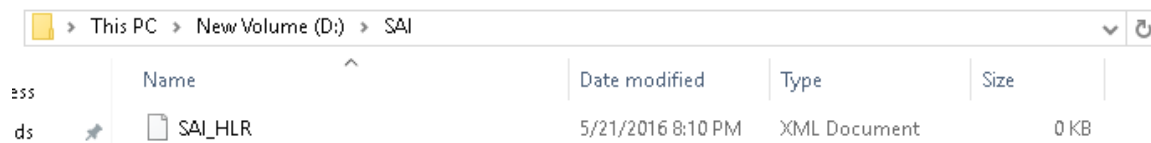
- Then it will prompt for the name for the decoder which you want to create.

For suppose I am giving the name of the decoder as **SAI_HLR** or **SAI_HLR.xml**

```
Enter the Name of the Decoder to generate:
SAI_HLR
```

Note: Decoder name should not contain space.

Then the SAI_HLR.xml file will be created in **D:\SAI** folder



| This PC > New Volume (D:) > SAI | | | | |
|---------------------------------|---------|-------------------|--------------|------|
| | Name | Date modified | Type | Size |
| ds | SAI_HLR | 5/21/2016 8:10 PM | XML Document | 0 KB |

- Then it will prompt for do you want Fingerprint to be added to your decoder.

If you want Mapping fingerprint enter **Y**

If you don't want Mapping fingerprint enter **N**, then only parser will be created without fingerprint.

```
Do you want Mapping Fingerprint to be Added to decoder(Y/N) :
Y
```

- If you enter **N**, then your parser will be created.

If you enter **Y**, then it will prompts for the Mapping fingerprint fields filename with path

For suppose my Fingerprint fields file is located in

C:\Users\battula.krishna\Desktop\ASCII_TO_DAIMOND\Samples\fingerprint.txt

```
Enter the filename containing the list of Mapping Fingerprint fields:
C:\Users\battula.krishna\Desktop\ASCII_TO_DAIMOND\Samples\fingerprint.txt
```

- Then it will prompt for the delimiter for the above fingerprint file.

For suppose my fingerprint file is having a pipe(|) delimiter.

```
Enter the field delimiter for above file (new line delimiter = \n):
|
```

- That's it now the decoder is generated.

```
Processing Done and Decoder Generated at D:\SAI\SAI_HLR.xml
```

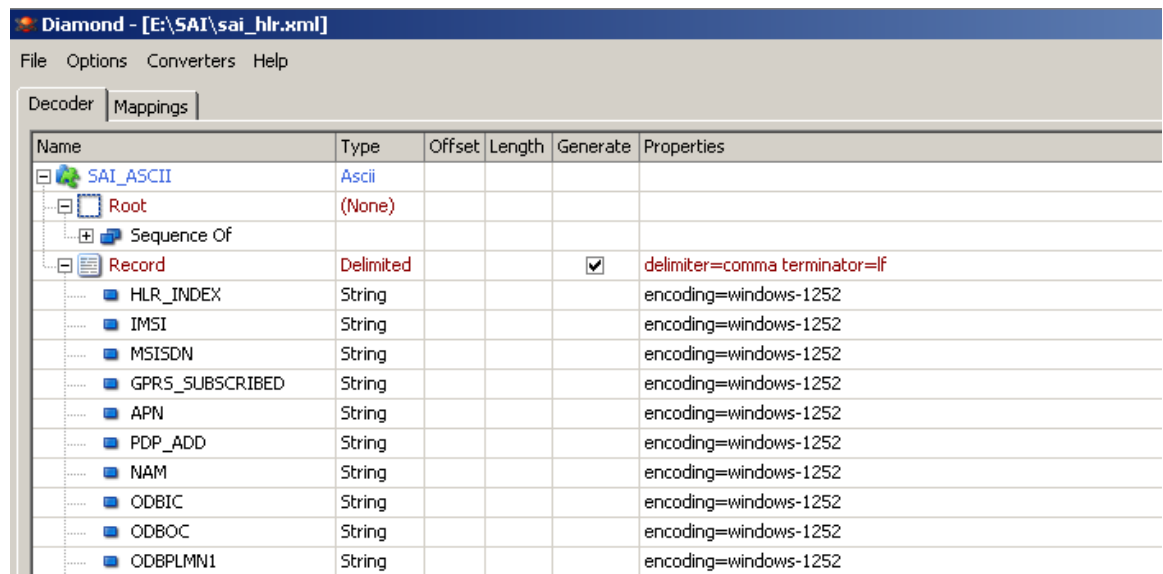
Sample Highlighted in green are the inputs that User has to give:


```

Enter the filename containing the list of parse input fields:
C:\Users\battula.krishna\Desktop\ASCII_TO_DAIMOND\Samples\ParseFields.txt
Enter the field delimiter for above file (new line delimiter = \n):
\n
Please enter Which type of delimited ASCII(comma,space,tab,|, etc.):
space
Enter the name of a direccctory to create in D:\ where output decoder Resides :
SAI
Enter the Name of the Decoder to generate:
SAI_HLR
Do you want Mapping Fingerprint to be Added to decoder(Y/N) :
Y
Enter the filename containing the list of Mapping Fingerprint fields:
C:\Users\battula.krishna\Desktop\ASCII_TO_DAIMOND\Samples\fingerprint.txt
Enter the field delimiter for above file (new line delimiter = \n):
|
Processing Done and Decoder Generated at D:\SAI\SAI_HLR.xml
  
```

2.3. Output

1. If the created decoder is opened in Diamond, then the below screen appears,



2. If we parse the sample data it is able to parse as below,

File Options Converters Help

Decoder Mappings

| Name | Type | Offset | Length | Generate | Properties |
|-----------------|-----------|--------|--------|-------------------------------------|-------------------------------|
| SAL_ASCII | Ascii | | | | |
| Root | (None) | | | | |
| Sequence Of | | | | | |
| Record | Delimited | | | <input checked="" type="checkbox"/> | delimiter=comma terminator=lf |
| HLR_INDEX | String | | | | encoding=windows-1252 |
| IMSI | String | | | | encoding=windows-1252 |
| MSISDN | String | | | | encoding=windows-1252 |
| GRPS_SUBSCRIBED | String | | | | encoding=windows-1252 |
| APN | String | | | | encoding=windows-1252 |
| PDP_ADD | String | | | | encoding=windows-1252 |
| NAM | String | | | | encoding=windows-1252 |
| ODBCIC | String | | | | encoding=windows-1252 |
| ODBOC | String | | | | encoding=windows-1252 |
| ODBPLMN1 | String | | | | encoding=windows-1252 |
| ODBPLMN2 | String | | | | encoding=windows-1252 |

Offset: 0 Length: 0

E:\Sites\MTN Swaziland\CR\H559860_1_20160505135807\H559860_1_20160505135807_0_data.csv (6.1 KB)

From: 0 To: 100 Parse + M

| Decoded | Parse Log | HLR_INDEX | IMSI | MSISDN | GRPS_SUBSCRIBED | APN | PDP_ADD | NAM | ODBCIC | ODBOC | ODBPLMN1 | ODBPLMN2 | ODBPLMN3 | ODBPLMN4 | ODBOAM | ODBCICT | ODBCICT2 | ODBINFO | ODBCICT3 | ODBCPOS | ODBCPOSTYPE | ODBCICF |
|---------|-----------|-----------------|-------------|--------|-----------------|-----|---------|-----|--------|-------|----------|----------|----------|----------|--------|---------|----------|---------|----------|---------|-------------|---------|
| 1 | Record 1 | 653104005817806 | 26878613182 | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Record 1 | 653104005820675 | 26878653418 | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | Record 1 | 653104005490431 | 26876934623 | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | Record 1 | 653104005817055 | 26878650485 | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | Record 1 | 653104900000016 | -1 | 0 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | Record 1 | 653104900000017 | -1 | 0 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Record 1 | 653104900000018 | -1 | 0 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

3. Also if you created the fingerprint you can see the same as below,

Output record type

Name:

| | | | |
|----|----------------|-------------------------------------|--------|
| 1 | fng_id | <input checked="" type="checkbox"/> | string |
| 2 | sfl_id | <input checked="" type="checkbox"/> | string |
| 3 | sts_id | <input checked="" type="checkbox"/> | string |
| 4 | record_address | <input checked="" type="checkbox"/> | string |
| 5 | record_length | <input checked="" type="checkbox"/> | string |
| 6 | record_number | <input checked="" type="checkbox"/> | string |
| 7 | record_type | <input checked="" type="checkbox"/> | string |
| 8 | hrl_filename | <input checked="" type="checkbox"/> | string |
| 9 | hrl_imsi | <input checked="" type="checkbox"/> | string |
| 10 | hrl_msisdn | <input checked="" type="checkbox"/> | string |
| 11 | hrl_services | <input checked="" type="checkbox"/> | string |
| 12 | hrl_ss | <input checked="" type="checkbox"/> | string |
| 13 | hrl_apn_id | <input checked="" type="checkbox"/> | string |
| 14 | hrl_calbinc | <input checked="" type="checkbox"/> | string |
| 15 | hrl_calboak | <input checked="" type="checkbox"/> | string |

OK Cancel

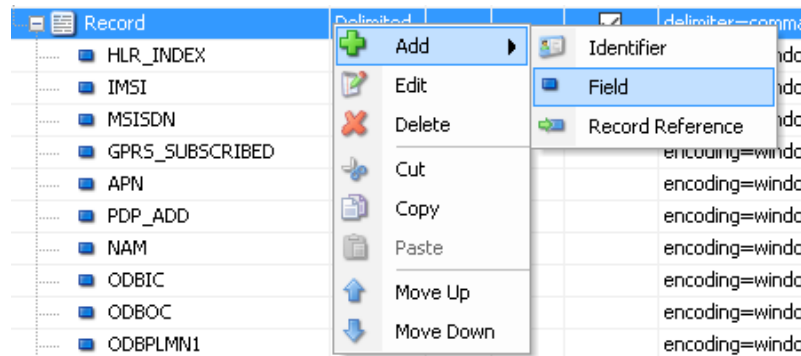
3. Efficiency

3.1. Normal Proof of Efficiency Improvement

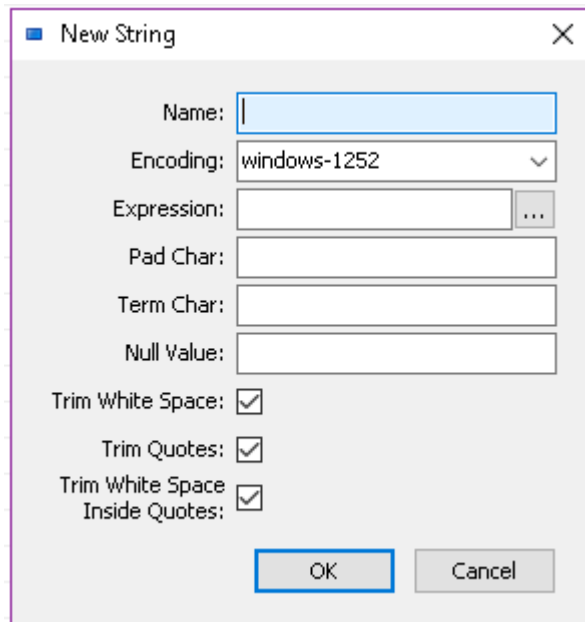
There are situations where the input parse fields will contain more no. of fields such as subscriber dumps (HLR, etc.).

If we want to create an ASCII decoder for an ASCII subscriber dump which is coming with input parse fields as more than 250, then in such cases creating a ASCII decoder will be a time consuming job which involves creating an each input parse field manually as below.

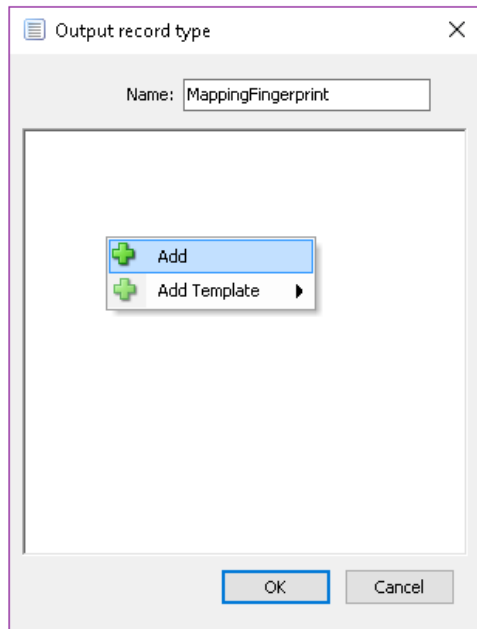
We have to add each field to the Record type.



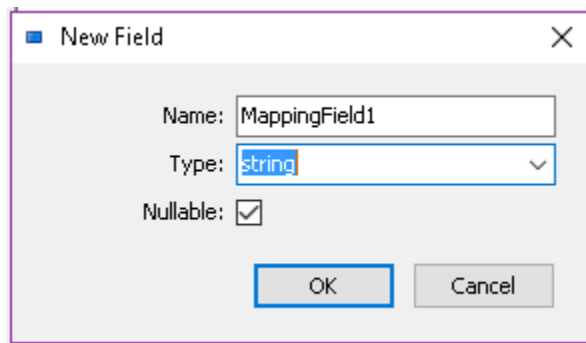
Then we need to type enter each name.



The same way for creating the finger print also.



And then entering each mapping field name,



The above process is a time consuming job. The time for the above process is based on no. of fields.

So if we have an ASCII to Diamond Converter then we can create an ASCII decoder within less than 5 mins which is a constant time for any no. of fields. So that we can save a lot of time with the this ASCII to Diamond Converter

3.2. Algorithmic Proof of Efficiency Improvement

3.2.1. Time Consumed by Manual Method

The manual creation of an ASCII decoder is based on the no. of parse fields and the no. of fingerprint fields.

For suppose we have **N** no. of parse fields.

And the no. of characters in each field will vary from field to field, which is a constantly varying variable **X**.

For suppose time taken for typing each character is **t** seconds. (**t** is a constant)

If 1st field contains **X1** character, then time taken by 1st field to enter is = **t.X1**

If 2nd field contains **X2** character, then time taken by 1st field to enter is = **t.X2**

If 3rd field contains **X3** character, then time taken by 1st field to enter is = **t.X3**

- - - - -
- - - - -
- - - - -
- - - - -

If Nth field contains **XN** character, then time taken by 1st field to enter is = **t.XN**

So Total time for making the parser is

$$T = t.X1 + t.X2 + t.X3 ++t.XN$$

$$T = N.t.(X1 + X2 + X3 ++ XN)$$

The expression $X1+X2+X3+.....XN$ which is varying based on the decoder to decoder and name of field to field, so the final $X1+X2+X3+.....XN$ is the total no of character used to create all the parse fields, let this be some **M** ($X1+X2+X3+.....XN = M$)

$$T = N.t.M$$

In above equation **t** is a time for typing each character which is a constant.

So the total time taken for creating a parser directly depends on **N & M** which is an order of $N*M$

$$T = \Theta(N*M) \text{ [Big O Notation]}$$

3.2.2. Time Consumed by ASCII to Diamond Converter

If we use the ASCII to Diamond Converter then the decoder can be developed in a constant time.

Time taken for making a parse fields file (Copying & Saving) = **a** (Constant time)

Executing the ASCII to Diamond Converter (Execution & entering input) = **b** (Constant time)

$$\text{Total time } T = a + b$$

$$T = c \text{ \textbackslash constant +constant =constant}$$

$$T = \Theta(c)$$

$$T = \Theta(1) \text{ which is a constant.}$$

$$\text{Hence } \Theta(1) < \Theta(N*M)$$

So time taken by ASCII to Diamond Converter is far less than time taken by manual creation.

Similarly for Mapping Finger print too.

4. Limitations:

1. It works for only Single Record type Decoder but not for multiple record types. (Which can be delivered as next version release)
2. It works for only delimited ASCII but not fixed length ASCII Decoder. (Which can be delivered as next version release)

5. Sample Files:

Parse Fields file:



ParseFields.txt

Fingerprint file:



fingerprint.txt

Corresponding Sample Data file:



HSS9860_1_2016050
5135807_0_data.csv