Vaxtor OCR Genesis

Grammar Rules Definition

V 1.0

November 2021

VAXTOR TECHNOLOGIES, Sector Foresta 1, 2º / 28760 Tres Cantos (Spain)
Telephone: +34 91 757 2211-- e-mail: info.emea@vaxtor.com - web: http://www.vaxtor.com

Introduction

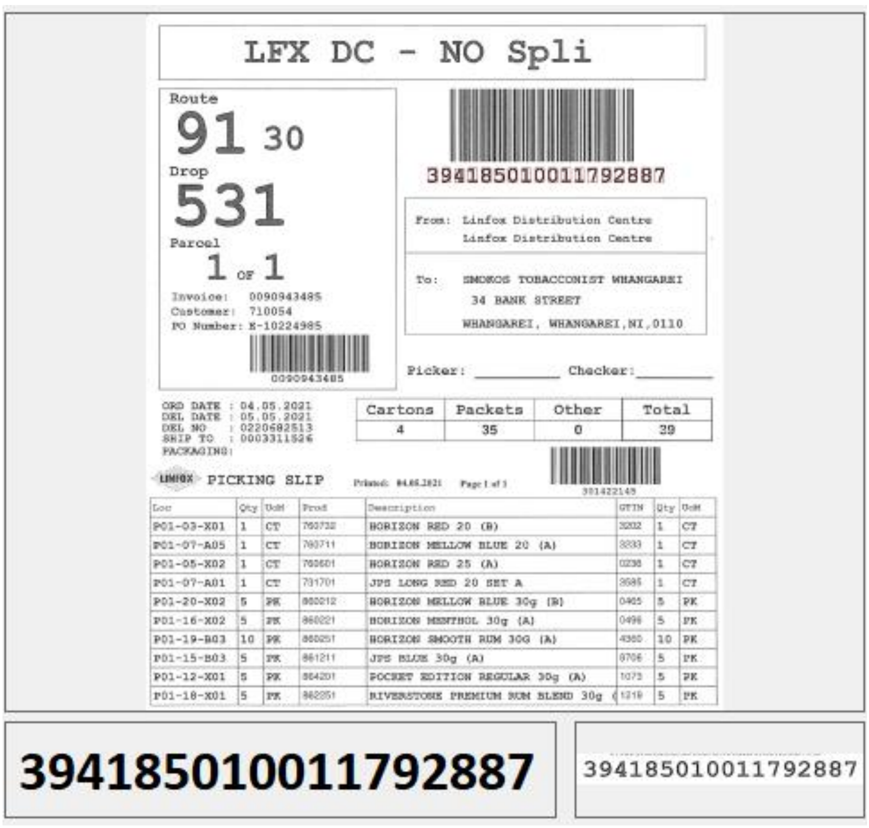
Vaxtor OCR Genesis is a software application developed by Vaxtor Technologies to read any combination of uppercase Latin characters and numbers arranged in up to three lines containing no (or small) blank spaces in between. It can operate under any kind of lighting conditions and is robust to the image quality, degradation, and font shape variations.

It is available under MS Windows or Linux PC platforms and is also available at the edge for various camera manufacturers. It can handle both video and still images and results can be sent in real time to third party end points using standard and open protocols such as: HTTP XML, JSON, ONVIF... amongst others.

The grammar definitions module

Vaxtor OCR Genesis searches and reads combinations of characters and/or numbers matching certain grammar or syntax rules, but it is also able to read open grammar structures.

The advantage of constraining the reads to certain grammars enables the user to exclude certain characters combinations that are not of interest and help achieve high levels of read accuracy.



LFX DC - NO Spli

Route
91 30
Drop
531
Parcel
1 OF 1

Invoice: 0090943485
Customer: 710054
PO Number: E-10224985
0090943485

394185010011792887

From: Linfox Distribution Centre
Linfox Distribution Centre

To: SMOKOS TOBACCONIST WHANGAREI
34 BANK STREET
WHANGAREI, WHANGAREI,NI,0110

Picker: __________ Checker: __________

ORD DATE : 04.05.2021
DEL DATE : 05.05.2021
DEL NO : 0220682513
SHIP TO : 0003311526
PACKAGING:

Cartons	Packets	Other	Total
4	35	0	39

LINFOX PICKING SLIP Printed: 04.05.2021 Page 1 of 1
301422145

Loc	Qty	UoM	Prod	Description	GTIN	Qty	UoM
P01-03-X01	1	CT	760732	HORIZON RED 20 (B)	3202	1	CT
P01-07-A05	1	CT	760711	HORIZON MELLOW BLUE 20 (A)	3233	1	CT
P01-05-X02	1	CT	760601	HORIZON RED 25 (A)	0238	1	CT
P01-07-A01	1	CT	731701	JPS LONG RED 20 SET A	3585	1	CT
P01-20-X02	5	PK	860212	HORIZON MELLOW BLUE 30g (B)	0465	5	PK
P01-16-X02	5	PK	860221	HORIZON MENTHOL 30g (A)	0496	5	PK
P01-19-B03	10	PK	860251	HORIZON SMOOTH RUM 30G (A)	4360	10	PK
P01-15-B03	5	PK	861211	JPS BLUE 30g (A)	0706	5	PK
P01-12-X01	5	PK	864201	POCKET EDITION REGULAR 30g (A)	1073	5	PK
P01-18-X01	5	PK	862251	RIVERSTONE PREMIUM RUM BLEND 30g (	1219	5	PK

394185010011792887

394185010011792887

The image above represents a good example of using grammar rules to focus the OCR read on a particular code, in this case an 18-digit number.

In addition to the grammar rules, we provide certain tools and filters to mark which codes we aim to read such as visual regions of interest, the range of the characters' heights in pixels, etc.

Grammar rules information structure

Vaxtor OCR Genesis requires a simple ASCII file which includes a set of syntactic and grammar rules to be applied, separated by a new line sequence.

The grammar file structure is made up 3 different sections:

- Character replacement and deletion rules
- Definition of variables
- Grammar rules

To define the components in each section we use three different types of items:

- Tokens: representing literals of a specific type
- Basic operators and definitions
- Characters and digits representing their own explicit value

Tokens, operators, and definitions

Replacement and deletion rules section

^	The rule for a replacement applies to the first characters of the sequence
$	The rule for a replacement applies to the last characters of the sequence
?	Mark character for deletion
~	Replacement rule separator

Definition of variables section

=	Variable assignment

Definition of variables and grammar rules sections

[,]	OR clause between the statements inside the brackets.

Grammar rules section

%{}	Variable usage
%0	Void, no character
N:	Rule matches if the character sequence is arranged in N lines where 1<=N<=3

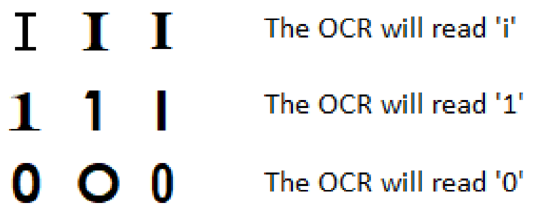
Common to all sections

%D	Numeric character: "0123456789"
%L	Alphabetic character: “ABCDEFGHIJKLMNOPQRSTUVWXYZ”
%V	Vowels “AEIOU”
%C	Consonants: “BCDFGHJKLMNPQRSTVWXYZ”
*	Alphanumeric character (numeric or alphabetic)
#	Start comment

Replacement rules section

Rules to erase or replace a specific alphanumeric with another one

- The OCR can’t differentiate O/0, it will always output 0 (zero)
- The OCR can’t differentiate I/1 when the representation is just a vertical line “I”



This section is particularly useful to “force” the replacements because the same alphanumeric shape can have different meaning depending on the context or use case.

Format:

[^]<input pattern>[$]~ [^][^]<output pattern>[$]

- The token around the pattern indicates the matching position in the sequence, in case of having no token the rule applies to the first match anywhere in the sequence.
- A pattern made up of literals and at least one explicit alphanumeric to be replaced or erased
- The tokens and the literals for the input pattern and the output pattern must be the same, only the explicit alphanumeric to replace or erase must change.
- The replacements and deletions apply sequentially when having more than one replacement or deletion statement.

Example 1:

Grammar rule of our text: 3 letters and 3 digits

Sample texts: ABC123, VHR394, PPW992...

Rule: If the first character is a 0, replace with O

^0%L~^O%L	=> Correct
^0%L%L%D%D%D~^O%L%L%D%D%D	=> Correct, more robust approach
0%L~O%L	=> Incorrect, it could replace the second character

Example 2:

Grammar rule of our text: one or more digits and 1 character (any)

Sample texts: 4364R, 449232S, 2530...

Rule: Replace the last 1 with I and 0 with O. We need 2 replacement rules:

%D1$~%DI$	=> If the last character is 1, replace with I
%D0$~%DO$	=> If the last character is 0, replace with O

Example 3:

Grammar rule of our text: one or more alphabetic characters

Sample texts: AR, PGKDS, GHJSO, SPII34...

Rule: Replace every 1 with I and every 0 with O. We need 2 replacement rules

1~I	=> Replace every ‘1’ with ‘I’
0~O	=> Replace every ‘0’ with ‘O’

Example 4:

Grammar rule of our text: 2 characters and 3 numbers

Sample texts: AO923, KJ145, OI992...

Rule: Replace any ‘1’ with ‘I’ and ‘0’ with ‘O’ in the first 2 positions

Valid but weak approach	Valid and robust approach
^1*~^I*	^1*%D%D%D~^I*%D%D%D
^0*~^O*	^0*%D%D%D ~^O*%D%D%D
^*1~^*I	^*1%D%D%D ~^*I%D%D%D
^*0~^*O	^*0%D%D%D ~^*O%D%D%D

Example 5:

Grammar rule of our text: 5 alphanumerics sequence

Sample texts: AB834, 98HUO, 1299W...

Rule: Replace any 5 alphanumeric sequence with the number 99999

^*****$~^99999$

Example 6:

Grammar rule of our text: 5 digits

Sample texts: 23425, 44536, 75433...

Rule: Remove every alphabetic character from the sequence

%L~?

%D%D%D%D%D

Text: 23456P	Result: 23456 (it matches the grammar)
Text: A82BC89U3	Result: 82893 (it matches the grammar)
Text: B8923CB	Result: 8923 (it does NOT match the grammar)

Grammar rules section

Set of grammar rules to follow by the characters sequence. The rules will be processed sequentially after executing the replacement section (if any).

Basic examples, fix length sequences

6 digits	%D%D%D%D%D%D
3 digits and 3 consonants	%D%D%D%C%C%C
Starts with 'P' followed by 5 digits	P%D%D%D%D%D
3 digits and 2 consonants	%D%D%D%C%C

Combining fix length sequences with replacements

3 digits and 2 characters

```
1*$~I*$                      # Replace last but one '1' with 'I'
0*$~O*$                      # Replace last but one '0' with 'O'
*1$~*I$                      # Replace last '1' with 'I'
*0$~*O$                      # Replace last '0' with 'O'
%D%D%D%L%L                   # Apply grammar rule, 3 digits and 2 characters
```

2 vowels and 4 digits assuming our 'I' letter is represented by 'I' or 6 digits sequence

```
^0*%D%D%D%D~^O*%D%D%D%D      # Replace the first '0' with 'O'
^*0%D%D%D%D~^*O%D%D%D%D      # Replace the second '0' with 'O'
%V%V%D%D%D%D                 # Apply rule 1
%D%D%D%D%D%D                 # Apply rule 2
```

Variable length sequences: introducing the 'OR' clause and the void literal

5-6 digits	%D%D%D%D%D[%D,%0]
5 digits ending in A, B, or C	%D%D%D%D%D[A,B,C]
2-4 digits and 1 or 2 consonants	%D%D[%D,%0] [%D,%0]%C[%C,%0]
Starting with 'AZ12' and 0-2 alphanumeric	AZ12[*,%0] [*,%0]

9 digits phone numbers starting with 609 with and without any 1-3 digits prefix

```
[%D,%0] [%D,%0] [%D,%0]609%D%D%D%D%D%D
```

3 or 4 digits numbers alone or preceded by 'AA' or 'BB'

```
[AA,BB,%0]%D%D%D[%D,%0]
```

Introducing the variables definition

The definition of variables appears in the second section, between the replacements and the grammar rules definition section. We should include those definitions before specifying any rule. The format is as follows:

<variable_name>=<grammar_rule>

The name of the variable includes alphanumeric and non-token characters, it may not include blanks.

3 digits and 2 consonants

```
MY_VARIABLE=%D%D%D                    # Variable definition
%{MY_VARIABLE}%C%C                    # Using the variable in a grammar rule
```

Combination of 2 characters AA,BB,CC,DD followed by 4 to 6 digits

```
COMBO=[AA,BB,CC,DD]
%{COMBO}%D%D%D%D[%D,%0] [%D,%0]
```

Note that any combination of characters ***not*** defined by COMBO won't match the rule

2 numbers plus AA,BB combination plus 2-4 numbers plus C,D,E,F,G characters

```
VAR1=[AA,BB]
VAR2=[C,D,E,F,G]
%D%D%{VAR1}%D%D[%D,%0] [%D,%0]%{VAR2}
```

Read the following words only: STOP, YIELD, SPEED, and 'speed' is followed by 2 or 3 digits

```
# Section replacement
^ST0P~^STOP                           # Replace the 0 with O
^Y1ELD~^YIELD                         # Replace the 1 with I

# Section variables definition
WORDS=[STOP,YIELD]

# Section grammar rules
%{WORDS}                              # Rule 1: stop or yield
SPEED%D%D[%D,%0]                      # Rule 2: speed plus a 2 or 3 digit number
```

Read numbers of 5-6 digits where every digit is equal or greater than 6

```
N=[6,7,8,9]                           # Numbers allowed
%{N}%{N}%{N}%{N}%{N}[%{N},%0]         # 5 or 6 digits number of allowed numbers only
```

Combination of 2 colors separated by an odd number

```
0~O
1~I
%L%L%LI%L%L%L~%L%L%L1%L%L%L

N=[1,3,5,7,9]
COLOR1=[RED,BLUE,YELLOW,ORANGE,BLACK,WHITE]
COLOR2=[GRAY,GREEN,PINK,BROWN,MAGENTA,VIOLET]

[%{COLOR1},%{COLOR2}]%{N}[%{COLOR1},%{COLOR2}]
```

<u>Selective rules depending on the number of lines the sequence is arranged</u>

The OCR can process 1, 2, or 3 line codes, the rules apply to any of them by default. However, it is possible to constrain the rules to the number of lines of the code.

<number_of_lines>:<grammar rule>	*# Number of lines can be 1, 2 or 3*
Rule: **%L%L%L%D%D%D%D**	# Default, it applies to any arrangement

ABC 1234	✓	ABC1234	✓

Rule: **1:%L%L%L%D%D%D%D**	# It applies to 1 line sequences only

ABC 1234	✗	ABC1234	✓

Rule: **2:%L%L%L%D%D%D%D**	# It applies to 2 lines sequences only

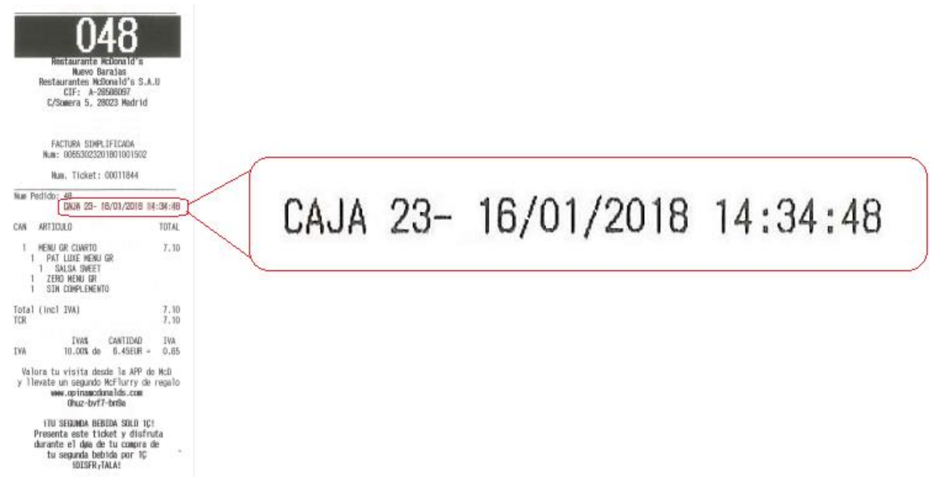
ABC 1234	✓	ABC1234	✗

Rule: **3:%L%L%L%D%D%D%D**	# It applies to 3 lines sequences only

ABC 1234	✗	ABC1234	✗

Example, extracting information from fast-food ticket restaurant

Real ticket



```
048
Restaurante McDonald's
Nuevo Barajas
Restaurantes McDonald's S.A.U
CIF:  A-28586097
C/Somera 5, 28023 Madrid

FACTURA SIMPLIFICADA
Num: 006530232018010001502

Num. Ticket: 00011844

Num Pedido: 48
        CAJA 23- 16/01/2018 14:34:48

CAN  ARTICULO                    TOTAL

 1   MENU GR CUARTO               7.10
   1   PAT LUXE MENU GR
     1   SALSA SWEET
   1   ZERO MENU GR
   1   SIN COMPLEMENTO

Total (incl IVA)                  7.10
TCR                               7.10

            IVA%     CANTIDAD     IVA
IVA       10.00% de  6.45EUR =   0.65

 Valora tu visita desde la APP de McD
y llevate un segundo McFlurry de regalo
       www.opinamcdonalds.com
           0huz-bvf7-bn9a

      ¡TU SEGUNDA BEBIDA SOLO 1€!
     Presenta este ticket y disfruta
     durante el dia de tu compra de
       tu segunda bebida por 1€
            ¡DISFRUTALA!
```

CAJA 23- 16/01/2018 14:34:48

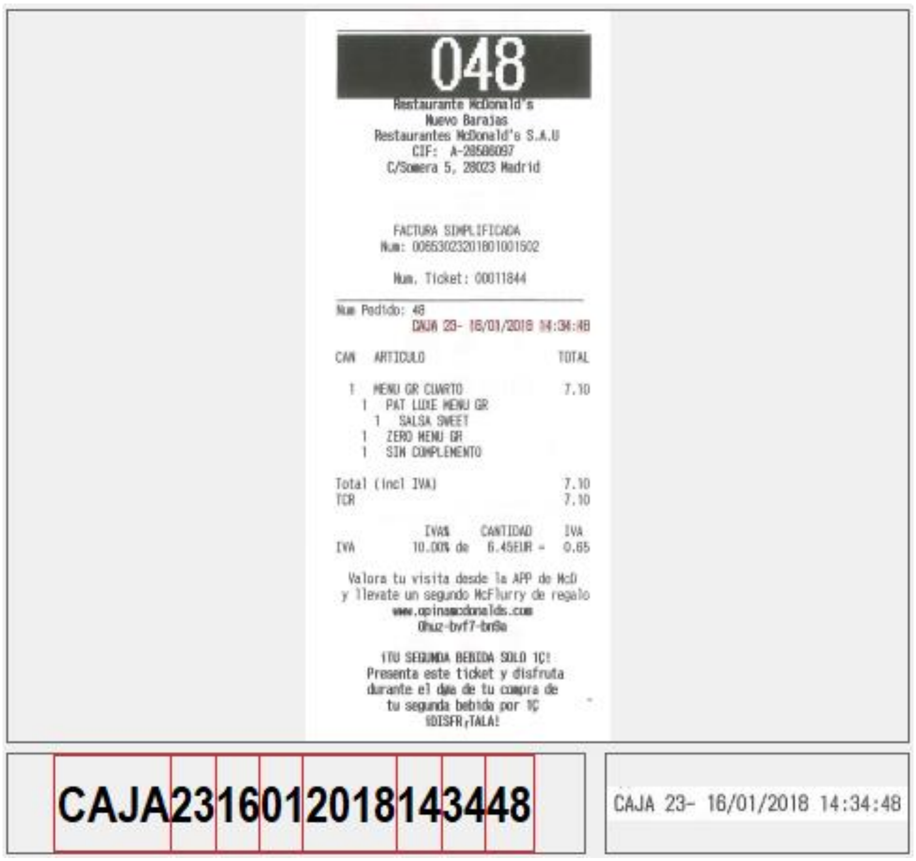
Grammar rules

DATE=%D%D%D%D%D%D%D%D

TIME=%D%D%D%D%D%D

CAJA%D[%D,%0]%{DATE}%{TIME}

OCR results



```
048
Restaurante McDonald's
Nuevo Barajas
Restaurantes McDonald's S.A.U
CIF:  A-28586097
C/Somera 5, 28023 Madrid

FACTURA SIMPLIFICADA
Num: 006530232018010001502

Num. Ticket: 00011844

Num Pedido: 48
        CAJA 23- 16/01/2018 14:34:48

CAN  ARTICULO                    TOTAL

 1   MENU GR CUARTO               7.10
   1   PAT LUXE MENU GR
     1   SALSA SWEET
   1   ZERO MENU GR
   1   SIN COMPLEMENTO

Total (incl IVA)                  7.10
TCR                               7.10

            IVA%     CANTIDAD     IVA
IVA       10.00% de  6.45EUR =   0.65

 Valora tu visita desde la APP de McD
y llevate un segundo McFlurry de regalo
       www.opinamcdonalds.com
           0huz-bvf7-bn9a

      ¡TU SEGUNDA BEBIDA SOLO 1€!
     Presenta este ticket y disfruta
     durante el dia de tu compra de
       tu segunda bebida por 1€
            ¡DISFRUTALA!
```

CAJA2316012018143448

CAJA 23- 16/01/2018 14:34:48