SFIMO-1775

GS1 element strings:

Application Data definition Format(AI & data)

identifier

01 GTIN N2+N14

10 Batch or lot number N2+X..20

11 Production date (YYMMDD) N2+N6

15 Best before date (YYMMDD) N2+N6

17 Expiration date (YYMMDD) N2+N6

21 Serial number N2+X..20

1D barcodes:

30 Count of items N1 <– variable length –> N8

Meaning of the format used:

N numeric digit

X alphanumeric characters

N2 fixed length of two numeric digits

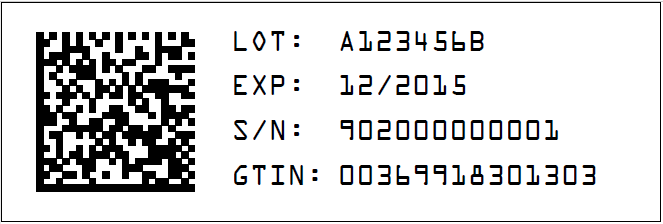
X...20 variable length with a maximum of 20 alphanumeric characters

(GTIN -> Global Trade Item Number)

Scanned labels (by file name):

GS1 DataMatrix

**Amring:**



Scanned chain:

**0100369918301303219020000000011715123110A123456B**

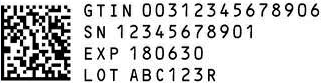
01 GTIN:00369918301303

21 S/N:902000000001

17 EXP:151231

10 LOT:A123456B

**Akorn:**



Scanned chain:

**010031234567890621123456789011718063010ABC123R**

01 GTIN:00312345678906

21 SN:12345678901

17 EXP:180630

10 LOT:ABC123R

**Actavis:**

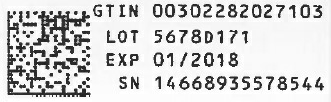
1D code:



Scanned chain:

301234567896

1)



Scanned chain:

**0100302282027103211466893557854417180128105678D171**

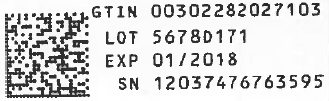
01 GTIN:00302282027103

21 SN:14668935578544

17 EXP:180128

10 LOT:5678D171

2)



Scanned chain:

**0100302282027103211203747676359517180128105678D171**

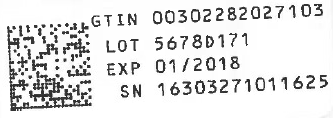
01 GTIN:00302282027103

21 SN:12037476763595

17 EXP:180128

10 LOT:5678D171

3)



Scanned chain:

**0100302282027103211630327101162517180128105678D171**

01 GTIN:00302282027103

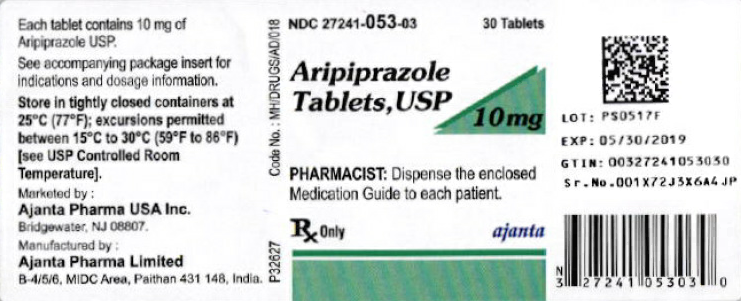
21 SN:16303271011625

17 EXP:180128

10 LOT:5678D171

**Ajanta**

1)



1D code chain:

327241053030

2D code chain:

**010032724105303021001X72J3X6A4JP1719053010PS0517F**

01 GTIN:00327241053030

21 SN:001X72J3X6A4JP 001X72J3D43DND (the other label)

17 EXP:190530

10 LOT:PS0517F

2)



Scanned chain:

**016032724105303221001X323347DJLD1719053010PS0517F**

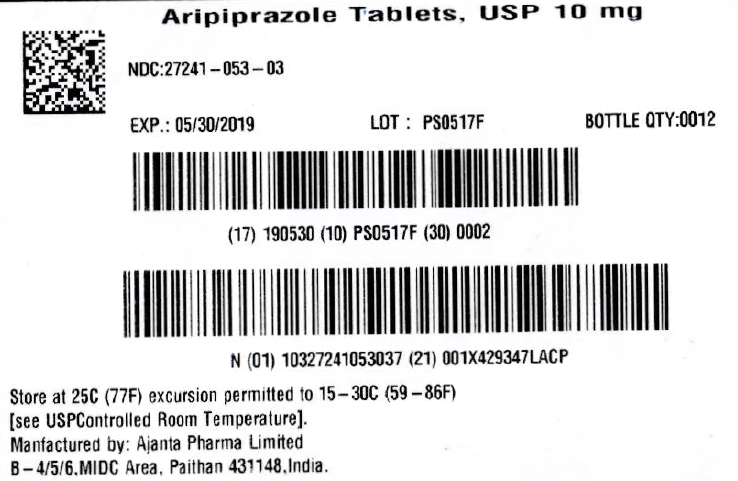
01 GTIN:60327241053032

21 SN:001X323347DJLD

17 EXP:190530

10 LOT:PS0517F

3)



Scanned chain:

**011032724105303721001X429347LACP1719053010PS0517F**

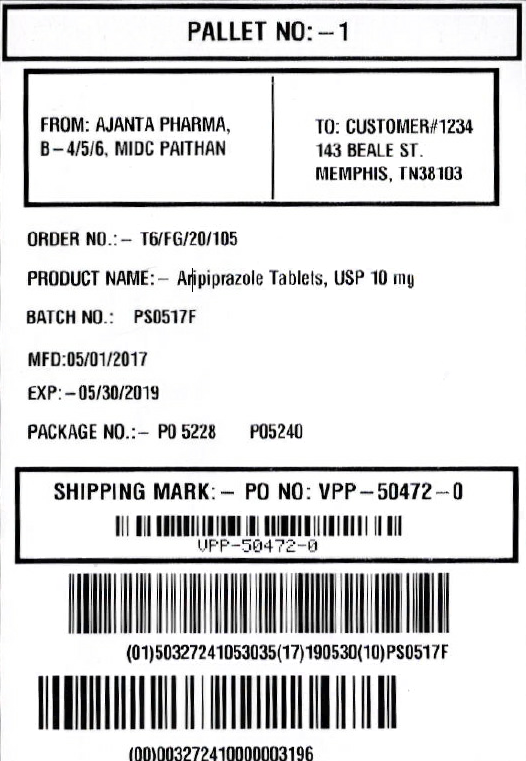
01 GTIN:10327241053037

21 SN:001X429347LACP

17 EXP:190530

10 LOT:PS0517F

4)



First barcode:

**VPP'50472'0**

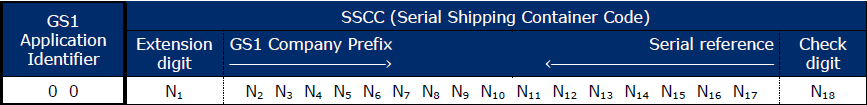
Second barcode:

Cannot read it

Third barcode:

**00003272410000003196**

The GS1 Application Identifier (00) indicates that the GS1 Application Identifier data field contains an SSCC (Serial Shipping Container Code). The SSCC is used to identify logistic units.



Concerns:

As I mentioned at the beginning of the first page there are identifiers assigned to each attribute, of which 4 have been identified as the most used (**code** – meaning – length).

**01** – GTIN – N2+N14

**21** – SN – N2+X..20 (after the identifier it could be since 1 to 20 characters)

**17** – EXP date – N2+N6

**10** – LOT – N2+X..20 (after the identifier it could be since 1 to 20 characters)

There are possibilities that S/N, EXP DATE and LOT attributes cannot be read correctly by JAVA because of the range length of characters for S/N and LOT attribute.

Example what I mean:

**Considering a JAVA approach for this.**

2D barcode (matrixData code)

**01**003**6991830130**3 **21**902000000001**17**151231 **10**A123456B

**1 2 3**

1. GTIN-14 attribute

Drug trade items that have an NDC can be given a GTIN that embeds the NDC. This is a best practice.

For an individual sale unit, the NDC is embedded in a GTIN-12 by combining the following elements in left-to-right order:

■ The digit “3”

■ The 10-digit NDC (on gray)

■ A check digit, computed from the previous 11 digits according to the GS1

For example, the 10-digit NDC 0001012345 results in GTIN-12 300010123455, which can be used to identify the individual sale unit for this NDC.

In some situations, the GTIN-12 has to be expressed in 14-digit format, which is done by adding two zero digits to the left of the GTIN-12. The 14-digit format of the previous example is 00300010123455. When storing a GTIN in a database, the 14-digit format should always be used.

There is no problem with this attribute because it will always have the same length.

2. SN and Exp. Date combination.

Due to SN value was read considering 20 positions (or characters) in the string when actually its value length is 12, we have a problem because the attributes SN and EXP Date were combined. The EXP Date (code and value) was considered as a part of SN value.

From Java I only read the string and I cannot know about how many characters will be read for SN value. The best solution for the trouble is stop the reading when next code is found, but there is another problem… what happen if exist the next numeric code value within the current string?

**01**00369918301303 **21**90200**17**00001 **17**151231 **10**A123456B

In the example above when I’m reading the SN attribute value (code 21), the reading will stop when it find the following code (17), this behavior would cut the SN attribute value, resulting in data inconsistency.

3. LOT value

This value can be affected as well by how the last values were read. It could happen something like the last example.

Let’s show it:

**01**00369918301303 **21**902000000001 **17**15**10**31 **10**A123456B

In the example above when I’m reading the EXP Date attribute value (code 17), the reading will stop when it find the following code (10 – LOT), in this case the algorithm will interpret that the month 10 (October) is the beginning of the next code (LOT), cutting the EXP Date and causing data inconsistency.

As well as the examples shown before, there are many combinations where the attribute code could be part of the value of another attribute. For JAVA is difficult to know the number of characters will be read (talking about SN and LOT attributes) because both of them have variable length. If the codes had the same length, it would be easy to know where I must stop read the string.

Also, it is worth mentioning that if the code had a special char before or after, such as “&”, “%”, “$”, etc., or perhaps if it was between parenthesis (Example: (01)00369918301303(21)902000000001…), the reading of the value would be much easier to read than try to guess or suppose the value.