

WCS Labels Changes

Aptean Ltd
Copyright © 2011-2026.

Contents

1 WCS Labels Changes.....	1
2 Latest versions.....	2
2.1 Groups.....	2
2.2 1.3 Barcodes.....	5
2.3 1.4 Optional Sections.....	6

1 WCS Labels Changes

The page is intended to provide a guideline on the construction and layout of the WCS Label formats.



2 Latest versions

```

<template>
<parameters version=?2?>
Carton : <item name="Carton" format="trim" />
Packing Date: <item name="SysDate" type=?date? format="DD/MMM/YYYY" />
Order : <item name="OrderNo" len="15" />
Customer: <item name="Customer" format="trim,ucase" />
<item name="AddrLine1" />
<item name="AddrLine2" />
<item name="Town" />
<item name="County" />
<item name="PostCode" />
<item name="Country" />
-----
Contents:
Stock          Batch          Qty
<group name="details" repeat="20">
<item name="Stock" len="15" /> <item name="Batch" len="8" /> <item name="Qty"
type=?numeric? format="####0" />
</group>
Label <item name="LabelNo" type=?numeric? format="####0" /> of <item name="BoxCount"
type=?numeric? format="####0" />
</template>

```

Version on TEMPLATE tag determines whether we are using old method with curly brackets or new method.

2.1 Groups

There is a requirement to use Deconsolidation for some promotional materials.

The process fits what they want to do for these items exactly, but this is only a small percentage of the stock being picked.

Need to check that Despatch Confirm cleans up Picking Container and Deconsolidation records for orders effectively, when using Final Media functionality.

Would require ?labels? produced from this process, which are more like mini packing lists:

```

Carton : XXXXXXXXXXXXXXXXXXXXXXXX
Packing Date: DD/MMM/YYYY

```



Order : XXXXXXXXXXXXXXXXXXXXXXXX

Customer: XXXXXXXXXXXXXXXXXXXXXXXX

Addr1

Addr2

Town

County

Country

Postcode

Contents:

Stock Batch Qty

XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXX XXXXX

<repeating up to X times>

Label X of Y

So we would produce as many labels as required to show the contents of the box, for the order.

Would need to extend label picking class so that you could specify sub groups, and how many times they could be printed.

```
<template>
```

```
<parameters version=?3?>
```

```
<text>Carton : </text><item name="Carton" format="trim" />
```

```
<text>Packing Date: </text><item name="SysDate" type=?date? format="DD/MMM/YYYY" />
```

```
<text>Order : </text><item name="OrderNo" len="15" />
```

```
<text>Customer: </text><item name="Customer" format="trim,ucase" />
```

```
<item name="AddrLine1" />
```

```
<item name="AddrLine2" />
```

```
<item name="Town" />
```

```
<item name="County" />
```

```
<item name="PostCode" />
```

```
<item name="Country" />
```

```
<text>-----</text>
```

```
<text>Contents:</text>
```

```
<text>Stock                   Batch           Qty</text>
```



```

<group name="details" repeat="20">
<item name="Stock" len="15" /> <item name="Batch" len="8" /> <item name="Qty"
type=?numeric? format="####0" />
</group>
<text>Label </text><item name="LabelNo" type=?numeric? format="####0" /><text> of
</text><item name="BoxCount" type=?numeric? format="####0" />
</template>

```

Need to design the tags fully, to decide what formatting details we can use.

Need MaxLen, Len, Format (Dates, Text and Numbers), mid and trim functions, etc.

Note: Have extended this to full XML, even the plain text bit.

Problem with this: if this isn't plain text, but printer language, each item print location is defined in the plain text. So:

```

^FO16,780^A0,30,26^FDProduct Code^FS
^FO310,780^A0,30,26^FDLot No^FS
^FO560,780^A0,30,26^FDctns^FS
^FO656,780^A0,30,26^FDPces^FS
^FO16,830^A0,30,26^FD12345678901234567890^FS
^FO310,830^A0,30,26^FD12345678901234^FS
^FO560,830^A0,30,26^FD00010^FS
^FO656,830^A0,30,26^FD00100^FS

```

This is ZPL to print some label headers, then print out the 4 items. The ?830? bit in the FO statement shows the y coordinate, where the line is printed. In our group, we need to specify the start (830) and the ?height? to be added to each line (lets say 80):

```

<template>
<parameters version=?3?>
<text>
^FO16,780^A0,30,26^FDProduct Code^FS
^FO310,780^A0,30,26^FDLot No^FS
^FO560,780^A0,30,26^FDctns^FS
^FO656,780^A0,30,26^FDPces^FS
</text>
<group name="details" repeat="5" ystart=?830? yspacing=?80?>
<text>^FO16,</text><item name=?ypos? /><text>^A0,30,26^FD</text><item name=?StockCode?
/><text>^FS</text>

```



```

<text>^FO310,</text><item name=?ypos? /><text>^A0,30,26^FD</text><item name=?Batch?
/><text>^FS</text>

<text>^FO560,</text><item name=?ypos? /><text>^A0,30,26^FD</text><item name=?Cases?
/><text>^FS</text>

<text>^FO656,</text><item name=?ypos? /><text>^A0,30,26^FD</text><item name=?Units?
/><text>^FS</text>

</group>

</template>

```

So, ystart defines the start y coordinate, yspacing defines how much to add on though each group, ypos defines the dictionary item which will be set by the previous 2 parameters.

This could also be expanded to xstart, xspacing and xpos, for producing lists. We would maybe also have to define the directionality - horizontal or vertical.

2.2 1.3 Barcodes

Need to include some information regarding barcode types. Most are just plain text producers e.g. CODE-39, but CODE-128 and EAN-128 allow changes based on content. For example:

12345678 can be codes as 128C

SSN12345678 can be coded as 128A or B, but the barcode is twice the length.

The barcode can be set to be in 128C, but the alpha portions can be set to be in 128A or B, with control sequences.

The language property on the parameters tag could control the characters used to change code type:

```
<parameters version=?3? language = ?ZPL?>
```

For example, a UCC barcode in zebra format:

```
^FD>;>802123456789012341501020310>6ROT45678>8>5370224^FS
```

So the format could be:

```

<text>^FD>;>802</text>

<item name=?StockCode? />

<text>15</text>

<item name=?SellBy? />

<text>10>6</text>

<item name=?CustBatch? />

<text>>8>537</text>

<item name=?Qty? fomat=?9999? />

<text>^FS</text>

```

That could cause problems to the parser though, so it might be better if:

```
<text>^FD</text>
```



```

<text type=?EAN128C?>02</text>
<item type=?EAN128C? name=?StockCode? />
<text>15</text>
<item name=?SellBy? />
<text>10</text>
<item type=?EAN128C? name=?CustBatch? />
<text>37</text>
<item name=?Qty? format=?9999? />
<text>^FS</text>

```

The property ?EAN128C? or ?CODE128C? on text tags will start the barcode ?>;? and set the code type C ?>8?

The property ?EAN128C? or ?CODE128C? on item tags will check the element for alphabetic characters and encapsulate the alpha text with ?>6? at the start and ?>8? at the end.

Still need some mechanism of terminating variable-length items though. Possibly send AI through as property, as follows:

```

<text>^FD</text>
<text type=?EAN128C? />
<item type=?EAN128C? ai=?02? name=?StockCode? />
<item type=?EAN128C? ai=?15? name=?SellBy? />
<item type=?EAN128C? ai=?10? name=?CustBatch? />
<item type=?EAN128C? ai=?37? name=?Qty? format=?9999? />
<text>^FS</text>

```

AI property would control putting the AI at the start of the printed field.

Could do same when checking for lowercase characters as well (change from B to A, etc).

Possibly would need a similar label printed from WMS (Pick Container Enquiry screen).

2.3 1.4 Optional Sections

I can see a need to modify this so that sections or items are only printed if certain criteria are met.

So:

```

<template version="3" language="ZPL">
<text>Carton : </text><item name="Carton" format="trim" />
<text>Packing Date: </text><item name="SysDate" type=?date? format="DD/MMM/YYYY" />
<text>Order : </text><item name="OrderNo" len="15" />
<text presentif=?customer?>Customer: </text>

```



```

<item name="Customer" format="trim,ucase" presentif=?customer? />
<item name="AddrLine1" />
<item name="AddrLine2" />
<item name="Town" />
<item name="County" />
<item name="PostCode" />
<item name="Country" />
<text>-----</text>
<text>Contents:</text>

```

etc

or

```

<template version="3" language="ZPL">
<text>Carton : </text><item name="Carton" format="trim" />
<text>Packing Date: </text><item name="SysDate" type=?date? format="DD/MMM/YYYY" />
<text>Order : </text><item name="OrderNo" len="15" />
<group name=?address? presentif=?customer?>
<text>Customer: </text>
<item name="Customer" format="trim,ucase" />
<item name="AddrLine1" />
<item name="AddrLine2" />
<item name="Town" />
<item name="County" />
<item name="PostCode" />
<item name="Country" />
</group>
<text>-----</text>
<text>Contents:</text>

```

etc

The `?presentif?` parameter controls whether the item or group contents should be displayed at all. Specifying a field name simply checks if there is a value in that string (or zero). Or you could specify a condition (`?qty<>0?`).

