

# How To Center a Code 128 Barcode Using Java Script

## Description

How to center a code 128 barcode, valid for code B, code C, and Function 1 characters, with the exception of "~".

## How to Center a Code128 Barcode using Java Script

Use the following Script, but replace 'CODE128' with the name of your data input field.

Align the width of the label is required. This can be a non-printing element and must be named BAR\_WIDTH

Function 1 character can be passed in the data using [FNC1].

If you will need to encode a "~" character in your data you will need to choose another ASCII character that will NOT be in your data to use as a place holder in the script.

## Article Number

2016059

## Versions

Loftware Label Manager version 9.5.3 or higher

## Environment

All Supported Loftware Print Server and Label Manager Environments

### Script

```
// Barcode Centering Script
// CENTER CODE 128 BARCODE
//
// L = (11C + 24)X
// where
// C = total number of characters
// X = x-dim/1000
// L = Length of symbol
// collect the necessary information
var xDim = String(label.fields.field.@name ==
'CODE128').barcode.@barLineXDim);
var labelWidth=Number(label.fields.field.@name
== 'BAR_WIDTH').@lineLength); // a nonprinting
line that is the width of the label
var barcodeData =
String(label.fields.field.@name ==
'CODE128').@data);
var func1 = label.fields.field.@name ==
"Func1").@data; // a field with formula data
source with only FNC1() in the formula
barcodeData =
barcodeData.replace(/\[FNC1\]/g,"~"); //
replace FNC1 calls in the data with ~ for
parsing
var barcode=barcodeData.split(""); // all
barcode data to array for analysis
var numcount=0;
var startCode="";
var barcodeL=0; //Number of barcode units
for (i=0;i<barcode.length;i++){
  if (isNaN(barcode[i])==false){
    numcount+=1;
    if (i==barcode.length-1){
      // last character is numeric
```

```

if (numcount==i-1){
  // barcode began with a string of numbers
  if (numcount%2==0){
    // even number - half the count
    barcodeL+=(numcount/2);
  }
  else{
    // odd number - half the count and one for
remaining number and one for startCode change
from C to B
    barcodeL+=Math.floor(numcount/2)+2;
  }
}
else if(numcount>=4){
  if (numcount%2==0){
    // even number - half the count add one
for startCode change B to C
    barcodeL+=(numcount/2)+1;
  }
  else{
    // odd number - half the count and one for
remaining number and for startCode change B to
C and one for startCode change C to B
    barcodeL+=Math.floor(numcount/2)+3;
  }
}
else if(numcount!=0){
  barcodeL+=numcount; // not enough
consecutive numeric values for startCode change
}
}
}

if (isNaN(barcode[i])==true){ // initiate
numeric compression calculation, if any, when
an non-numeric character is encountered
  // if character is
  if (barcode[i]=="~" && numcount%2==0){
    barcodeL+=1; // increment count for character
    if (i==0){
      barcodeL+=1; // incrment count for initial
start UCC code.
    }
  }
}
else{
  if (numcount==i-1){
    // barcode began with a string of numbers
    if (numcount%2==0){
      // even number - half the count add 1 for
starCode change C to B
      barcodeL+=(numcount/2)+1;
    }else{
      // odd number - half the count and one for

```

```

remaining number and for startCode change C to
B
    barcodeL+=Math.floor(numcount/2)+2;
    }
    }
    else if(numcount>=4){
        if (numcount%2==0){
            // even number - half the count add one
for startCode change B to C and startCode
change C to B
            barcodeL+=(numcount/2)+2;
        }
        else{
            // odd number - half the count and one for
remaining number and for startCode change B to
C and startCode change C to B
            barcodeL+=Math.floor(numcount/2)+3;
        }
    }
    else if(numcount!=0){
        barcodeL+=numcount; // not enough
consecutive numeric values for startCode change
    }
        barcodeL+=1;// increment count for character
        numcount=0; // reset numeric count to 0
    }
    }
}
var actualLength=Number(11*barcodeL); // number
of bars per unit
actualLength = Number(actualLength +24); //
number of bars for Start/Stop/CheckDigit
actualLength = Number(actualLength * xDim); //
actual width of barcode in 1/1000"
actualLength = Number(actualLength / 1000); //
width of barcode in inches
var newStartPoint = Number((labelWidth -
actualLength) / 2) ; // compute new left edge
position of barcode
newStartPoint =
Math.round(newStartPoint*Math.pow(10,2))/Math.p
ow(10,2); // round value to nearest 1/1000
label.fields.field.(@name
=='CODE128').@posLeft= newStartPoint; // move
barcode
// replace "~" with FNCl()

```

```
barcodeData = barcodeData.replace(/~/g,func1);  
label.fields.field.(@name  
=='CODE128').@data=barcodeData;
```

**Note**

This may not function correctly if Autodiscriminate is enabled (Zebra printers only)

## Related Articles

[Page:Cannot Print Greater-Than \(">"\) Characters in Code-128 Barcode When Using Zebra Printer](#)

[Page:Encoding a Keystroke Character Within A Code 39 Symbology](#)

[Page:Code 39 or Code 39 \(Full ASCII\) Barcode Yields Unexpected Results When Scanned](#)

[Page:How To Center a Code 128 Barcode Using Java Script](#)

[Page:How To Center a Code 39 Full ASCII Barcode Using Java Script](#)