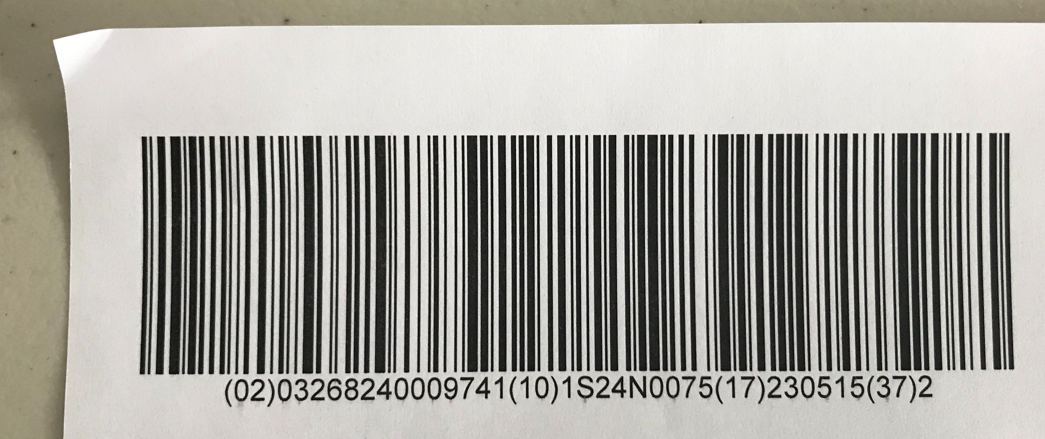
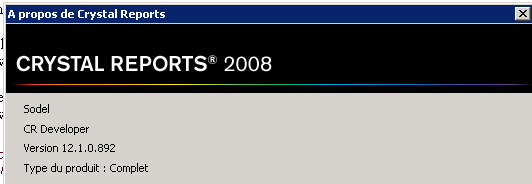


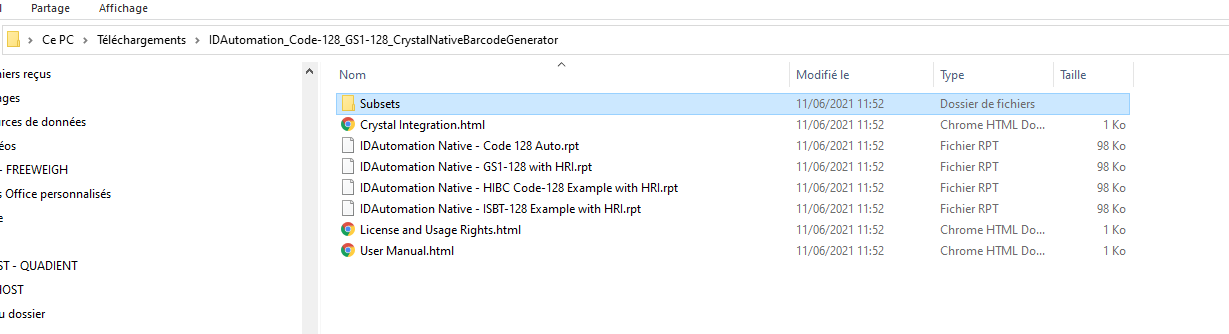
Nous avons besoin de cette codification



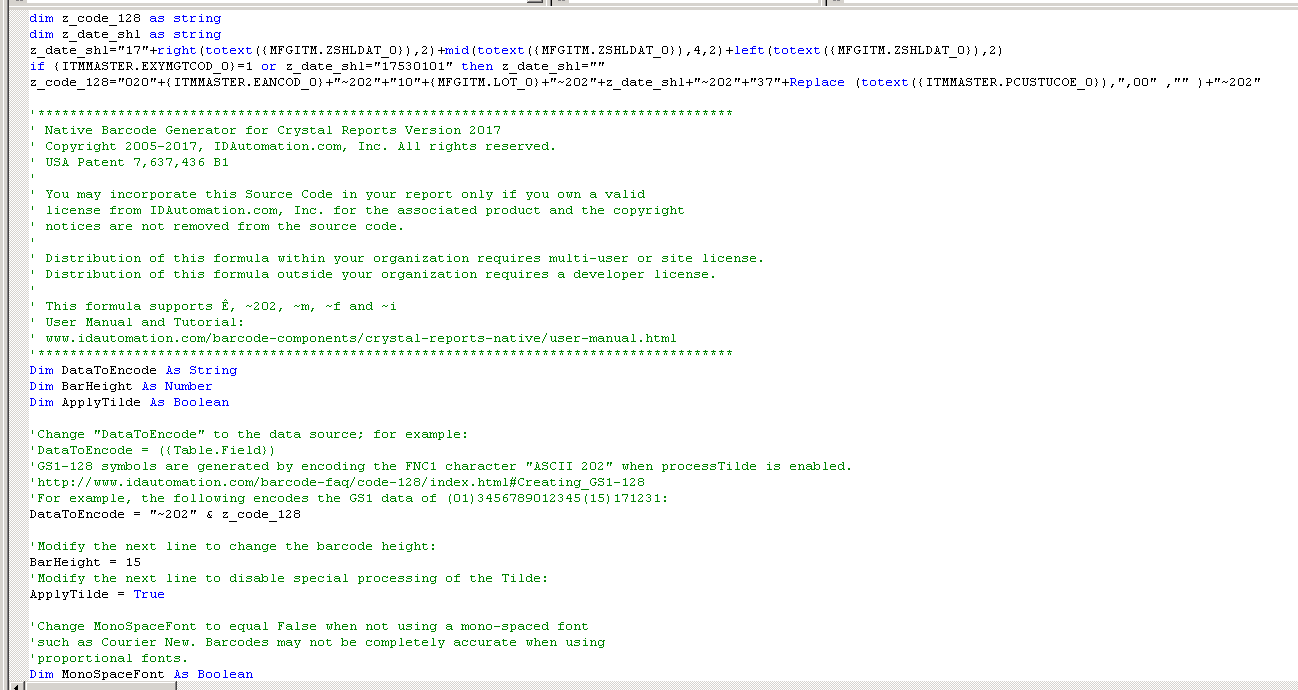
Notre version



Je n’ai pas les polices GS1 128 (Fonts) dans mon package



Dans Crystal :



dim z\_code\_128 as string

dim z\_date\_shl as string

z\_date\_shl="17"+right(totext({MFGITM.ZSHLDAT\_0}),2)+mid(totext({MFGITM.ZSHLDAT\_0}),4,2)+left(totext({MFGITM.ZSHLDAT\_0}),2)

if {ITMMASTER.EXYMGTCOD\_0}=1 or z\_date\_shl="17530101" then z\_date\_shl=""

z\_code\_128="020"+{ITMMASTER.EANCOD\_0}+"~202"+"10"+{MFGITM.LOT\_0}+"~202"+z\_date\_shl+"~202"+"37"+Replace (totext({ITMMASTER.PCUSTUCOE\_0}),",00" ,"" )+"~202"

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Native Barcode Generator for Crystal Reports Version 2017

' Copyright 2005-2017, IDAutomation.com, Inc. All rights reserved.

' USA Patent 7,637,436 B1

'

' You may incorporate this Source Code in your report only if you own a valid

' license from IDAutomation.com, Inc. for the associated product and the copyright

' notices are not removed from the source code.

'

' Distribution of this formula within your organization requires multi-user or site license.

' Distribution of this formula outside your organization requires a developer license.

'

' This formula supports Ê, ~202, ~m, ~f and ~i

' User Manual and Tutorial:

' www.idautomation.com/barcode-components/crystal-reports-native/user-manual.html

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim DataToEncode As String

Dim BarHeight As Number

Dim ApplyTilde As Boolean

'Change "DataToEncode" to the data source; for example:

'DataToEncode = ({Table.Field})

'GS1-128 symbols are generated by encoding the FNC1 character "ASCII 202" when processTilde is enabled.

'http://www.idautomation.com/barcode-faq/code-128/index.html#Creating\_GS1-128

'For example, the following encodes the GS1 data of (01)3456789012345(15)171231:

DataToEncode = "~202" & z\_code\_128

'Modify the next line to change the barcode height:

BarHeight = 15

'Modify the next line to disable special processing of the Tilde:

ApplyTilde = True

'Change MonoSpaceFont to equal False when not using a mono-spaced font

'such as Courier New. Barcodes may not be completely accurate when using

'proportional fonts.

Dim MonoSpaceFont As Boolean

MonoSpaceFont = True

Dim DataToPrint as String

Dim WeightedTotal As Number

Dim SymbolString As String

Dim StringLength As Number

StringLength = Len(DataToEncode)

Dim CurrentCharNum As Number

Dim CurrentValue As Number

Dim WeightValue As Number

Dim NumCharsTaken As Number

Dim CheckDigitValue As Number

Dim PrintableString As String

Dim Factor As Number

Dim I As Number

Dim J As Number

Dim CorrectFNC As Number

Dim nwPattern As String

Dim OnlyCorrectData As String

Dim OnlyNumberData As String

Dim OnlyNumberDataRev As String

Dim DataToFormat As String

Dim C128Start As String

PrintableString = ""

Formula = ""

Dim ArrayBase As Number

Dim CurrentChar As String

ArrayBase = 1 'Array Base is 1 in Crystal Reports and 0 in Microsoft VB

Dim CurrentEncoding As String

CurrentEncoding = ""

DataToPrint = ""

CorrectFNC = 0

'in case ApplyTilde is null, set it to false

If ApplyTilde <> True Then ApplyTilde = False

PrintableString = ""

'ApplyTilde ~???

If ApplyTilde Then

Dim OutString As String

Dim CharsAdded As Number

Dim StringToCheck As String

Dim M10StringLength As Number

Dim M10OnlyCorrectData As String

Dim M10Factor As Number

Dim M10WeightedTotal As Number

Dim M10CheckDigit As Number

Dim strTemp as String

Dim datamod as String

Dim F43Set as String

Dim F43WeightTotal as Number

Dim M37Correct as String

Dim M37CharCode As Number

Dim M37CharValue As Number

Dim M37WeightTotal As Number

Dim M37CheckDigit As Number

F43Set = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ-. $/+%"

OnlyCorrectData = ""

StringLength = Len(DataToEncode)

For I = 1 To StringLength

If (I < StringLength - 2) And Mid(DataToEncode, I, 1) = "~" And IsNumeric(Mid(DataToEncode, I + 1, 3)) Then

CurrentCharNum = Val(Mid(DataToEncode, I + 1, 3))

OnlyCorrectData = OnlyCorrectData & ChrW(CurrentCharNum)

I = I + 3

Else

OnlyCorrectData = OnlyCorrectData & Mid(DataToEncode, I, 1)

End If

Next I

DataToEncode = OnlyCorrectData

End If

'ApplyTilde ~m??

If ApplyTilde Then

OutString = ""

StringLength = Len(DataToEncode)

For I = 1 To StringLength

'' LM 2017

'' check tilde string for ~mNN (MOD10 for GS1) or ~fNN (MOD43 for HIBC) or ~iNN (MOD37 for ISBT)

If (i < StringLength - 2) And \_

(Mid(DataToEncode, i, 2) = "~m" Or Mid(DataToEncode, i, 2) = "~f" Or Mid(DataToEncode, i, 2) = "~i") And \_

IsNumeric(Mid(DataToEncode, i + 2, 2)) Then

WeightValue = Val(Mid(DataToEncode, i + 2, 2))

If Mid(DataToEncode, i, 2) = "~m" Then

'' ~mNN (MOD10 for GS1)

datamod = ""

CharsAdded = 0

For j = i-1 To 1 Step -1

If IsNumeric(Mid(OutString, j, 1)) Then

datamod = Mid(OutString, j, 1) & datamod

CharsAdded = CharsAdded + 1

End If

'when the number of digits added to StringToCheck equals the weight value exit the for loop

If CharsAdded = WeightValue Then

Exit For

End If

Next j

'' MODU10

M10OnlyCorrectData = ""

M10StringLength = Len(datamod)

'Check to make sure data is numeric and remove dashes, etc.

For j = 1 To M10StringLength

'Add all numbers to OnlyCorrectData string

if IsNumeric(Mid(datamod, j, 1)) Then M10OnlyCorrectData = M10OnlyCorrectData & Mid(datamod, j, 1)

Next j

'Generate MOD 10 check digit

M10Factor = 3

M10WeightedTotal = 0

M10StringLength = Len(M10OnlyCorrectData)

For j = M10StringLength To 1 Step -1

'Get the value of each number starting at the end

'CurrentCharNum = Mid(M10NumberData, I, 1)

'Multiply by the weighting factor which is 3,1,3,1...

'and add the sum together

M10WeightedTotal = M10WeightedTotal + (Val(Mid(M10OnlyCorrectData, j, 1)) \* M10Factor)

'Change factor for next calculation

M10Factor = 4 - M10Factor

Next j

'Find the CheckDigit by finding the smallest number that = a multiple of 10

j = (M10WeightedTotal Mod 10)

If j <> 0 Then

CheckDigitValue = (10 - j)

Else

CheckDigitValue = 0

End If

OutString = OutString & ChrW(CheckDigitValue + 48)

elseif Mid(DataToEncode, i, 2) = "~f" Then

'' ~fNN (MOD43 for HIBC)

F43WeightTotal = 0

datamod = ""

CharsAdded = 0

For j = i-1 To 1 Step -1

If InStr(F43Set, Mid(OutString, j, 1)) > 0 Then

datamod = Mid(OutString, j, 1) & datamod

CharsAdded = CharsAdded + 1

End If

'when the number of digits added to StringToCheck equals the weight value exit the for loop

If CharsAdded = WeightValue Then

Exit For

End If

Next j

M10StringLength = Len(datamod)

For j = 1 To M10StringLength

F43WeightTotal = F43WeightTotal + (InStr(F43Set, Mid(datamod, j, 1)) - 1)

Next j

F43WeightTotal = (F43WeightTotal Mod 43)

OutString = OutString & Mid(F43Set, F43WeightTotal + 1, 1)

elseif (Mid(DataToEncode, i, 2) = "~i") Then

'' ~iNN (MOD37 for ISBT)

M37Correct = ""

M37CharCode = 0

M37CharValue = 0

M37WeightTotal = 0

M37CheckDigit = 0

datamod = ""

CharsAdded = 0

For j = i-1 To 1 Step -1

M37CharCode = Asc(Mid(OutString, j, 1))

If (M37CharCode > 47 And M37CharCode < 58) Or (M37CharCode > 64 And M37CharCode < 91) Then

datamod = Mid(OutString, j, 1) & datamod

CharsAdded = CharsAdded + 1

End If

'when the number of digits added to StringToCheck equals the weight value exit the for loop

If CharsAdded = WeightValue Then

Exit For

End If

Next j

M10StringLength = Len(datamod)

For j = 1 To M10StringLength

M37CharCode = Asc(Mid(datamod, j, 1))

'' 0-9

If (M37CharCode > 47 And M37CharCode < 58) Then M37CharValue = M37CharCode - 48

'' A-Z

If (M37CharCode > 64 And M37CharCode < 591) Then M37CharValue = M37CharCode - 55

M37WeightTotal = ((M37WeightTotal + M37CharValue) \* 2) Mod 37

Next j

M37CheckDigit = (38 - M37WeightTotal) Mod 37

strTemp = ""

if (M37CheckDigit < 10) then strTemp = Chr(M37CheckDigit + 48) '' 0-9

if (M37CheckDigit < 36 And M37CheckDigit > 9) Then strTemp = Chr(M37CheckDigit + 55) '' A-Z

if (M37CheckDigit = 36) Then strTemp = Chr(42)

OutString = OutString & strTemp

end if

i = i + 3

Else

OutString = OutString & Mid(DataToEncode, I, 1)

End If

Next I

DataToEncode = OutString

End If

DataToFormat = DataToEncode

OnlyCorrectData = ""

DataToEncode = ""

DataToPrint = ""

Dim SetC128(128) As String

SetC128 = Array( \_

"EFF", "FEF", "FFE", "BBG", "BCF", "CBF", "BFC", "BGB", "CFB", "FBC", \_

"FCB", "GBB", "AFJ", "BEJ", "BFI", "AJF", "BIF", "BJE", "FJA", "FAJ", \_

"FBI", "EJB", "FIB", "IEI", "IBF", "JAF", "JBE", "IFB", "JEB", "JFA", \_

"EEG", "EGE", "GEE", "ACG", "CAG", "CCE", "AGC", "CEC", "CGA", "ECC", \_

"GAC", "GCA", "AEK", "AGI", "CEI", "AIG", "AKE", "CIE", "IIE", "ECI", \_

"GAI", "EIC", "EKA", "EII", "IAG", "ICE", "KAE", "IEC", "IGA", "KEA", \_

"IMA", "FDA", "OAA", "ABH", "ADF", "BAH", "BDE", "DAF", "DBE", "AFD", \_

"AHB", "BED", "BHA", "DEB", "DFA", "HBA", "FAD", "MIA", "HAB", "CMA", \_

"ABN", "BAN", "BBM", "ANB", "BMB", "BNA", "MBB", "NAB", "NBA", "EEM", \_

"EME", "MEE", "AAO", "ACM", "CAM", "AMC", "AOA", "MAC", "MCA", "AIM", \_

"AMI", "IAM", "MAI", "EDB", "EBD", "EBJ")

'Here we select character set A, B or C for the START character

'Start A = "EDB"

'Start B = "EBD"

'Start C = "EBJ"

StringLength = Len(DataToFormat)

CurrentCharNum = AscW(Mid(DataToFormat, 1, 1))

'Set A?

If CurrentCharNum < 32 Then C128Start = "EDB"

'Set B?

If CurrentCharNum > 31 And CurrentCharNum < 127 Then C128Start = "EBD"

If CurrentCharNum = 197 Then C128Start = "EBD"

'Updated V5.08 by BDA

'Set C?

If ((StringLength > 3) And IsNumeric(Mid(DataToFormat, 1, 4))) Then C128Start = "EBJ"

'202 & 212-215 is for the FNC1, with this Start C is mandatory

If CurrentCharNum = 202 Then C128Start = "EBJ"

If CurrentCharNum > 211 Then C128Start = "EBJ"

If C128Start = "EDB" Then CurrentEncoding = "A"

If C128Start = "EBD" Then CurrentEncoding = "B"

If C128Start = "EBJ" Then CurrentEncoding = "C"

For I = 1 To StringLength

CurrentCharNum = AscW(Mid(DataToFormat, I, 1))

'check for FNC1 in any set which is ASCII 202 and ASCII 212-219

If ((CurrentCharNum > 201) And (CurrentCharNum < 219)) Then

DataToEncode = DataToEncode & ChrW(202)

'check for switching to character set C

ElseIf CurrentCharNum = 197 Then

If CurrentEncoding = "C" Then 'switch to B

DataToEncode = DataToEncode & ChrW(200)

CurrentEncoding = "B"

End If

DataToEncode = DataToEncode & ChrW(197)

ElseIf ((I < StringLength - 2) And (IsNumeric(Mid(DataToFormat, I, 1))) And (IsNumeric(Mid(DataToFormat, I + 1, 1))) And (IsNumeric(Mid(DataToFormat, I, 4)))) Or ((I < StringLength) And (IsNumeric(Mid(DataToFormat, I, 1))) And (IsNumeric(Mid(DataToFormat, I + 1, 1))) And (CurrentEncoding = "C")) Then

'check to see if we have an odd number of numbers to encode,

'if so, stay in current set and then switch to save space

If CurrentEncoding <> "C" Then

J = I

Factor = 3

Do While J <= StringLength And IsNumeric(Mid(DataToFormat, J, 1))

Factor = 4 - Factor

J = J + 1

Loop

If Factor = 1 Then

'if so stay in current set for 1 character to save space

DataToEncode = DataToEncode & ChrW(CurrentCharNum)

I = I + 1

End If

End If

'switch to set C if not already in it

If CurrentEncoding <> "C" Then DataToEncode = DataToEncode & ChrW(199)

CurrentEncoding = "C"

CurrentChar = Mid(DataToFormat, I, 2)

CurrentValue = Val(CurrentChar)

'set the CurrentValue to the number of String CurrentChar

DataToEncode = DataToEncode & ChrW(CurrentValue + 32)

I = I + 1

'check for switching to character set A

ElseIf (I <= StringLength) And ((AscW(Mid(DataToFormat, I, 1)) < 31) Or ((CurrentEncoding = "A") And (AscW(Mid(DataToFormat, I, 1)) > 32 And (AscW(Mid(DataToFormat, I, 1))) < 96))) Then

'switch to set A if not already in it

If CurrentEncoding <> "A" Then DataToEncode = DataToEncode & ChrW(201)

CurrentEncoding = "A"

'Get the ASCII value of the next character

CurrentCharNum = AscW(Mid(DataToFormat, I, 1))

If CurrentCharNum < 32 Then

DataToEncode = DataToEncode & ChrW(CurrentCharNum + 96)

ElseIf CurrentCharNum > 32 Then

DataToEncode = DataToEncode & ChrW(CurrentCharNum)

End If

'check for switching to character set B

ElseIf (I <= StringLength) And (((AscW(Mid(DataToFormat, I, 1))) > 31) And ((AscW(Mid(DataToFormat, I, 1)))) < 127) Then

'switch to set B if not already in it

If CurrentEncoding <> "B" Then DataToEncode = DataToEncode & ChrW(200)

CurrentEncoding = "B"

'Get the ASCII value of the next character

CurrentCharNum = AscW(Mid(DataToFormat, I, 1))

DataToEncode = DataToEncode & ChrW(CurrentCharNum)

End If

Next I

'<<< Calculate Modulo 103 Check Digit >>>

If C128Start = "EDB" Then WeightedTotal = 103 'CurrentEncoding = "A"

If C128Start = "EBD" Then WeightedTotal = 104 'CurrentEncoding = "B"

If C128Start = "EBJ" Then WeightedTotal = 105 'CurrentEncoding = "C"

StringLength = Len(DataToEncode)

For I = 1 To StringLength

CurrentCharNum = AscW(Mid(DataToEncode, I, 1))

If CurrentCharNum < 135 Then CurrentValue = CurrentCharNum - 32

If CurrentCharNum > 134 Then CurrentValue = CurrentCharNum - 100

If CurrentCharNum = 194 Then CurrentValue = 0

PrintableString = PrintableString & SetC128(CurrentValue + ArrayBase)

CurrentValue = CurrentValue \* I

WeightedTotal = WeightedTotal + CurrentValue

Next I

CheckDigitValue = (WeightedTotal Mod 103)

DataToEncode = ""

'GIAH produces the stop character.

DataToPrint = C128Start & PrintableString & SetC128(CheckDigitValue + ArrayBase) & "GIAH"

'Universal to SymbolString conversion

'At this point in the code, DataToPrint is the entire barcode string for the Universal Font

SymbolString = ""

StringLength = Len(DataToPrint)

For I = 1 To StringLength

'Get the value of each

CurrentCharNum = AscW(Mid(DataToPrint, I, 1))

If CurrentCharNum >= 65 And CurrentCharNum <= 68 Then SymbolString = SymbolString & Cstr((11+(CurrentCharNum - 65)),0,"")

If CurrentCharNum >= 69 And CurrentCharNum <= 72 Then SymbolString = SymbolString & Cstr((21+(CurrentCharNum - 69)),0,"")

If CurrentCharNum >= 73 And CurrentCharNum <= 76 Then SymbolString = SymbolString & Cstr((31+(CurrentCharNum - 73)),0,"")

If CurrentCharNum >= 77 And CurrentCharNum <= 80 Then SymbolString = SymbolString & Cstr((41+(CurrentCharNum - 77)),0,"")

If CurrentCharNum >= 87 And CurrentCharNum <= 88 Then SymbolString = SymbolString & Cstr((11+(CurrentCharNum - 87)),0,"")

If CurrentCharNum >= 89 And CurrentCharNum <= 90 Then SymbolString = SymbolString & Cstr((21+(CurrentCharNum - 89)),0,"")

Next I

'SymbolString to Native Conversion

'At this point in the code, SymbolString is the entire barcode string in number form

'For example: 211214142112214121221114311321211232

'Convert SymbolString to PrintableString

Dim ww As String

Dim wb As String

Dim bw As String

Dim bb As String

ww = ChrW(32) 'SPACE

If MonoSpaceFont = False Then ww = ChrW(32) & ChrW(32) & ChrW(32)

wb = ChrW(9616)

bw = ChrW(9612)

bb = ChrW(9608)

CurrentValue = 0

PrintableString = ""

Dim NextDigitUsed As Number

Dim StartOver As Number

NextDigitUsed = 0

Factor = 3

Dim SymbolStringLength

SymbolStringLength = Len(SymbolString) - 1

For I = 1 To SymbolStringLength

CurrentValue = Val(Mid(SymbolString, I, 1))

If NextDigitUsed = 1 Then

'Because the next digit is used, remove 1 from CurrentValue

'unless CurrentValue is already 1

If CurrentValue > 1 then

CurrentValue = CurrentValue - 1

NextDigitUsed = 0

Else

'CurrentValue was used in a previous step

NextDigitUsed = 1

End If

End If

If NextDigitUsed = 1 And CurrentValue = 1 Then

'There is nothing more here to do because the CurrentValue

'was used in a previous step

NextDigitUsed = 0

Else

For J = 1 to CurrentValue

If J = CurrentValue Then

NextDigitUsed = 1

'We are at the end of the bar segment, borrow 1 segment from the next

If Factor = 3 Then PrintableString = PrintableString & bw

If Factor = 1 Then PrintableString = PrintableString & wb

Else

NextDigitUsed = 0

If Factor = 3 Then PrintableString = PrintableString & bb

If Factor = 1 Then PrintableString = PrintableString & ww

J = J+1

End If

Next J

End If

Factor = 4 - Factor

Next I

For I = 1 to BarHeight

Formula = Formula & ChrW(10) & ChrW(13) & PrintableString

Next I