




Id. de artículo: 65323 - Última revisión: martes, 16 de agosto de 2005 - Versión: 2.1

## Programa básico para leer COBOL BCD (COMP-3) números

 Haga clic aquí para ver en paralelo el artículo de traducción automática y el artículo original en inglés.  Advertencia: Artículo de Traducción Automática, vea la extensión de responsabilidad.  Renuncia a responsabilidad de los contenidos de la KB sobre productos a los que ya no se ofrece asistencia alguna

### Resumen

A continuación es un programa Basic que lee un archivo de datos creado por Microsoft COBOL que contengan números (COMP-3) a decimal codificado binario. Para ello, lee en el archivo de un byte en un momento, interpreta esos bytes el formato de decimal codificado binario (BCD) y realiza cálculos basados en esas interpretaciones para generar equivalente números de punto flotante de precisión doble básica.

Esta información se aplica a las versiones de Microsoft QuickBasic 4.00, 4.00b y 4.50 para MS-DOS; para las versiones de Microsoft Basic Compiler 6.00 y 6.00b para MS-DOS y MS OS/2; y a las versiones de sistema de desarrollo de Professional básica de Microsoft (PDS) 7.00 y 7.10 para MS-DOS y MS OS/2.

La información sobre números de decimal codificado binario y el formato de COMP-3 se aplica a las versiones de compilador de Microsoft COBOL 3.00 y 3.00a para MS-DOS y MS OS/2 para MS-DOS y MS OS/2 y sistema de desarrollo de Microsoft COBOL Professional versión 4.00.

### Más información

COBOL 3.00 3.00a y 4.00 de Microsoft admite un tipo de datos comúnmente como formato de decimal codificado binario (BCD). El nombre sintáctico de este tipo de datos en COBOL es 3 COMP.

Los bytes que utiliza para almacenar un número 3 COMP pueden contener dos dígitos de base 10 cada. El alto trozo (4 bits) almacena un dígito del número y el bajo trozo (4 bits) almacena el siguiente dígito a la derecha. Es decir, el trozo alto almacena un dígito cuyo lugar en el número tiene una potencia de 10 que es mayor que el dígito en el trozo bajo uno. Además, el primer byte utilizado para almacenar el número contendrá el dígito con el mayor potencia de 10; bytes posteriores contendrá dígitos que tienen sucesivamente menor potencias de 10. (Nota: hay dos nibbles en todos los bytes.)

Se realiza un caso especial para el último byte, donde el trozo alto almacena el último dígito del número y el trozo bajo almacena el signo del número. El trozo bajo contendrá 15 si el número es sin signo, 12 para positivo y 13 para negativo. Además, si el número de nibbles necesarios para dígitos más el trozo de inicio de sesión no rellenará un número de bytes, el alto trozo del byte almacenar el primer dígito se establece en 0.

Como ejemplo, aquí es una representación binaria de cómo se almacenará el número de 69 (con signo positivo):

```
+----- First Byte -----+----- Second Byte -----+ | | +- High Nibble +- Low Nibble --+ | | | |
+-----+-----+-----+-----+ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
+-----+-----+-----+-----+ | | | | +----- 0 -----+----- 6 -----+----- 9 -----+ Positive ----+
```

abajo es un programa Basic (BCDTODBL.BAS) que convierte los números de COMP-3 en un archivo de datos de COBOL a números de precisión doble de Basic. El programa lee cada byte del archivo se cuecialmente, interpretar como un par de dígitos o un dígito y un signo. El programa sabe que ha leído todos los dígitos en un número cuando el trozo bajo del último byte leído contiene un valor de signo (12, 13 ó 15). Cuando esto ocurre, el programa circula a través de todos los dígitos acumulados para el número y la multiplica por su potencia de 10 asociado. El resultado de este cálculo iterativo está asignado a un número de punto flotante de precisión doble. Por último, se realiza un cálculo más el número poder contendrá la cantidad de decimales es especificado por el usuario.

Siguiendo el programa Basic es un programa de COBOL (WRITEBCD.CBL) que crea un archivo que contiene una cantidad especificada por el usuario de COMP-3 firmado números. Para compilar y vincular el programa en COBOL 3.00 o 3.00a, escriba los siguientes comandos desde el símbolo de MS-DOS u OS/2:

```
cobol writebcd; link writebcd;
```

las líneas de compilación para la versión 4.00 de COBOL Professional de desarrollo del sistema (PDS) son:

```
cobol writebcd; link writebcd,,,coblib cobapi;
```

básica el programa se puede ejecutar o compilar desde dentro de los entornos QB.EXE o QBX.EXE. Consulte la documentación básica para obtener información sobre el uso de los entornos QB.EXE o QBX.EXE.

### BCDTODBL.BAS

```
DIM Digits%(15) 'Holds the digits for each number (max = 16). DIM BasicEqv%(1000) 'Holds the Basic equivalent of each COMP-3 number. 'Clear the screen, get the
filename and the amount of decimal places 'desired for each number, and open the file for sequential input: CLS INPUT "Enter the COBOL data file name: ", FileName$
INPUT "Enter the number of decimal places desired: ", Decimal% OPEN FileName$ FOR INPUT AS #1 DO UNTIL EOF(1) 'Loop until the end of the file is reached. Byte$ =
INPUT$(1, 1) 'Read in a byte. IF Byte$ = CHR$(0) THEN 'Check if byte is 0 (ASC won't work on 0). Digits%(HighPower%) = 0 'Make next two digits 0. Increment Digits%
(HighPower% + 1) = 0 'the high power to reflect the HighPower% = HighPower% + 2 'number of digits in the number 'plus 1. ELSE HighNibble% = ASC(Byte$) \ 16
'Extract the high and low LowNibble% = ASC(Byte$) AND &HF 'nibbles from the byte. The Digits%(HighPower%) = HighNibble% 'high nibble will always be a 'digit. IF
```

```

LowNibble% <= 9 THEN 'If low nibble is a 'digit, assign it and Digits%(HighPower% + 1) = LowNibble% 'increment the high HighPower% = HighPower% + 2 'power
accordingly. ELSE HighPower% = HighPower% + 1 'Low nibble was not a digit but a Digit% = 0 '+' or - signals end of number. 'Start at the highest power of 10 for the
number and multiply 'each digit by the power of 10 place it occupies. FOR Power% = (HighPower% - 1) TO 0 STEP -1 Basicq#(E%)=Basicq#(E%) + (Digits%(Digit%) *
(10^Power%)) Digit% = Digit% + 1 NEXT 'If the sign read was negative, make the number negative. IF LowNibble% = 13 THEN Basicq#(E%) = Basicq#(E%) - (2 *
Basicq#(E%)) END IF 'Give the number the desired amount of decimal places, print 'the number, increment E% to point to the next number to be 'converted, and
reinitialize the highest power. Basicq#(E%) = Basicq#(E%) / (10 ^ Decimal%) PRINT Basicq#(E%) E% = E% + 1 HighPower% = 0 END IF END IF LOOP CLOSE 'Close the
COBOL data file, and end.

```

## WRIT EBCD.CBL

```

$SET ANS85 IDENTIFICATION DIVISION. PROGRAM-ID. WriteBCD. ENVIRONMENT DIVISION. INPUT-OUTPUT SECTION. FILE-CONTROL. SELECT BCDfile ASSIGN TO "BCD.DAT". DATA
DIVISION. FILE SECTION. FD BCDfile. 01 BCDNumber PIC S9(5) COMP-3. WORKING-STORAGE SECTION. * Holds the number of COMP-3 items to write to the file. 01
TotalNumbers PIC 99 COMP-5. PROCEDURE DIVISION. * Get the number of COMP-3 items to write to file. DISPLAY "Enter the number of items to write:". ACCEPT
TotalNumbers. * Open the COMP-3 file for sequential output and write each * number out to the file as the user enters it. OPEN OUTPUT BCDfile. PERFORM TotalNumbers
TIMES ACCEPT BCDNumber WRITE BCDNumber END-PERFORM. * Close the COMP-3 file and end. CLOSE BCDfile.

```

### La información de este artículo se refiere a:

- Microsoft QuickBasic 4.0
- Microsoft QuickBASIC 4.0b
- Microsoft QuickBasic 4.5 para MS-DOS
- Microsoft BASIC Compiler 6.0
- Microsoft BASIC Compiler 6.0b
- Microsoft BASIC Professional Development System 7.0
- Microsoft BASIC Professional Development System 7.1

**Palabras clave:** kbnt KB65323 kbMts



**Traducción automática** IMPORTANTE: Este artículo ha sido traducido por un software de traducción

automática de Microsoft (<http://support.microsoft.com/gp/mtdetails>) en lugar de un traductor humano. Microsoft le ofrece artículos traducidos por un traductor humano y artículos traducidos automáticamente para que tenga acceso en su propio idioma a todos los artículos de nuestra base de conocimientos (Knowledge Base). Sin embargo, los artículos traducidos automáticamente pueden contener errores en el vocabulario, la sintaxis o la gramática, como los que un extranjero podría cometer al hablar el idioma. Microsoft no se hace responsable de cualquier imprecisión, error o daño ocasionado por una mala traducción del contenido o como consecuencia de su utilización por nuestros clientes. Microsoft suele actualizar el software de traducción frecuentemente. Haga clic aquí para ver el artículo original (en inglés): [65323](http://support.microsoft.com/kb/65323/en-us/)

(<http://support.microsoft.com/kb/65323/en-us/>)



**Renuncia a responsabilidad de los contenidos de la KB sobre productos a los que ya no se ofrece asistencia alguna** El presente artículo se escribió para productos para los que Microsoft ya no ofrece soporte técnico. Por tanto, el presente artículo se ofrece "tal cual" y no se rá actualizado.



**¿Necesita más ayuda?**

Contactar con un profesional de soporte técnico por correo electrónico, online o por teléfono

Soporte Microsoft

**Microsoft**

©2010 Microsoft