

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

[MÁS INFORMACIÓN](#) [ENTENDIDO](#)



Consultorio COBOL DB2 JCL REXX ...Y MAS PRINCIPIANTES El equipo Padrinos Contacto LIBROS

 Buscar

MIÉRCOLES, 23 DE MARZO DE 2011

Ejemplo 4: generando un listado.

Un programa típico que nos encontraremos en cualquier aplicación es aquel que genera un listado. ¿Qué a qué nos referimos con listado? Mejor verlo para hacernos una idea:

```
-----+---1---+---2---+---3---+---4---+
LISTADO DE EJEMPLO DEL CONSULTORIO COBOL
DIA: 23-03-11          PAGINA: 1
-----
NOMBRE      APELLIDO
-----
ANA        LOPEZ
ANA        MARTINEZ
ANA        PEREZ
ANA        RODRIGUEZ

TOTAL ANA   : 04
```

LISTADO DE EJEMPLO DEL CONSULTORIO COBOL
DIA: 23-03-11 PAGINA: 2

```
-----+---1---+---2---+
NOMBRE      APELLIDO
-----
BEATRIZ    GARCIA
BEATRIZ    MOREDA
BEATRIZ    OTERO

TOTAL BEATRIZ : 03
TOTAL NOMBRES: 07
```

Aquí vemos un listado de 2 páginas. Ambas páginas tienen una parte común que se denomina cabecera y que, por lo general, será la misma en todas las páginas. Suele contener un título que describa al listado y la fecha en que ha sido generado. Lo único que cambia es el número de página en el que estamos[^]

Después tenemos una "subcabecera" también común en cada página, en este caso la subcabecera incluye a "NOMBRE" y "APELLIDO".

En nuestro listado de ejemplo hemos querido que cada nombre salga en una página distinta. Al final de cada página de nombre escribimos un "subtotal" con el número de registros que hemos escrito para ese nombre.

Al final del listado escribiremos una línea de "totales" con el total de registros escritos.

El fichero de partida para este programa será el siguiente:

```
-----+---1---+---2
ANA      LOPEZ
ANA      MARTINEZ
ANA      PEREZ
ANA      RODRIGUEZ
BEATRIZ  GARCIA
BEATRIZ  MOREDA
BEATRIZ  OTERO
```

SIGUENOS EN INTERNET



Como siempre: lo urgente no deja tiempo para lo importante. - Mafalda

AMIGOS DEL CONSULTORIO

ÁFRICA ARAGÓN MOLLÁ
Formadora en COBOL-PL/I-CICS-DB2

www.universocobol.com

COBOLEROS

COMPARTE CONOCIMIENTO

Envía tu artículo, noticia o manual relacionado con nuestro mundillo y si es elegido lo publicaremos en el consultorio.

consultoriocobol@gmail.com



APADRINA UN PICADOR

[Donar](#)

Ir a Padrinos

MANUALES

- Principiantes COBOL
- Principiantes PL/I

ZONA DESCARGAS

Descárgate nuestra App para Android



LISTADOS DE ERRORES

- Errores de JCL
- Errores DB2
- Errores de ficheros
- Errores de sistema
- Errores IMS
- Errores CICS
- Oncodes PL/I

ARCHIVO DEL BLOG

marzo 2011 (17)

249 readers
BY FEEDBURNER

RECIBE LAS ACTUALIZACIONES DEL CONSULTORIO!

Subscríbete al Consultorio Cobol por Email

SEGUIDORES

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

[MÁS INFORMACIÓN](#) [ENTENDIDO](#)

```
-----+---1-----+---2
ANA      LOPEZ
BEATRIZ  MOREDA
ANA      PEREZ
BEATRIZ  OTERO
ANA      RODRIGUEZ
BEATRIZ  GARCIA
ANA      MARTINEZ
```



[Seguir](#)

JCL:

```
//***** BORRADO *****
//BORRADO EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEL FICHERO.NOMBRES.APELLIDO.ORDENADO
DEL FICHERO.CON.LISTADO
SET MAXCC = 0
//***** ORDENAMOS EL FICHERO POR NOMBRE Y APELLIDO *****
//SORT01 EXEC PGM=SORT
//SORTIN   DD DSN=FICHERO.NOMBRES.APELLIDO,DISP=SHR
//SORTOUT  DD DSN=FICHERO.NOMBRES.APELLIDO.ORDENADO,
//           DISP=(,CATLG),SPACE=(TRK,(50,10))
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
      SORT FIELDS=(1,9,CH,A,10,10,CH,A)
//***** EJECUCION DEL PROGRAMA PRUEBA3 *****
//PROG4 EXEC PGM=PRUEBA4
//SYSOUT  DD SYSOUT=*
//ENTRADA DD DSN=FICHERO.NOMBRES.APELLIDO.ORDENADO,DISP=SHR
//SALIDA  DD DSN=FICHERO.CON.LISTADO,
//           DISP=(NEW, CATLG, DELETE),SPACE=(TRK,(50,10)),
//           DCB=(RECFM=FBA,LRECL=133)
/*
```

En este JCL tenemos 3 pasos:

Paso 1: Borrado de ficheros que se generan durante la ejecución, visto ya en otros ejemplos.

Paso 2: Ordenación del fichero de entrada usando el SORT. Toda la información sobre el SORT la tenéis en [SORT vol.1: SORT, INCLUDE](#).

Paso 3: Ejecución del programa que genera el listado. Tenemos como fichero de entrada el fichero de salida del SORT, y como fichero de salida ojito: indicaremos RECFM=FBA siempre para listados. Esto significa que el fichero contiene caracteres ASA, que son los que le indican a la impresora los saltos de línea que tiene que hacer al imprimir. Lo iremos viendo con el programa de ejemplo. La longitud del fichero(LRECL) suele ser 133, debido a que se imprimen en hojas A4 en formato apaisado.

Programa:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PRUEBA3.
*=====
* PROGRAMA QUE LEE DE FICHERO Y ESCRIBE EN FICHERO
*=====
*
ENVIRONMENT DIVISION.
*
CONFIGURATION SECTION.
*
SPECIAL-NAMES.
DECIMAL-POINT IS COMMA.
*
INPUT-OUTPUT SECTION.
```

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN ENTENDIDO

```
----- STATUS IS FS-LISTADO.  
*  
DATA DIVISION.  
*  
FILE SECTION.  
*  
* FICHERO DE ENTRADA DE LONGITUD FIJA (F) IGUAL A 20.  
FD ENTRADA RECORDING MODE IS F  
BLOCK CONTAINS 0 RECORDS  
RECORD CONTAINS 20 CHARACTERS.  
01 REG-ENTRADA PIC X(20).  
*  
* FICHERO DE LISTADO DE LONGITUD FIJA (F) IGUAL A 132.  
FD LISTADO RECORDING MODE IS F  
BLOCK CONTAINS 0 RECORDS  
RECORD CONTAINS 132 CHARACTERS.  
01 REG-LISTADO PIC X(132).  
*  
WORKING-STORAGE SECTION.  
*  
* FILE STATUS  
*  
01 FS-STATUS.  
05 FS-ENTRADA PIC X(2).  
88 FS-ENTRADA-OK VALUE '00'.  
88 FS-ENTRADA-EOF VALUE '10'.  
05 FS-LISTADO PIC X(2).  
88 FS-LISTADO-OK VALUE '00'.  
*  
* SWITCHES  
*  
01 WB-FIN-ENTRADA PIC X(1) VALUE 'N'.  
88 FIN-ENTRADA VALUE 'S'.  
*  
* CONTADORES  
*  
01 WC-LINEAS PIC 9(2).  
01 WC-NOMBRES PIC 9(2).  
01 WC-TOTALES PIC 9(2).  
*  
* VARIABLES  
*  
01 WX-REGISTRO-ENTRADA.  
05 WX-NOMBRE PIC X(9).  
05 WX-APELLIDO PIC X(10).  
  
01 WX-NOMBRE-ANT PIC X(9).  
  
01 WX-FEC-DDDDMAA.  
05 WX-FEC-DD PIC 9(2).  
05 FILLER PIC X VALUE '-'.  
05 WX-FEC-MM PIC 9(2).  
05 FILLER PIC X VALUE '-'.  
05 WX-FEC-AA PIC 9(2).  
  
01 WX-FECHA PIC 9(6).  
*  
* REGISTRO LISTADO  
*  
01 CABECERA1.  
05 FILLER PIC X(40)  
VALUE 'LISTADO DE EJEMPLO DEL CONSULTORIO  
- 'COBOL'.  
01 CABECERA2.  
05 FILLER PIC X(5) VALUE 'DIA: '.  
05 LT-FECHA PIC X(8).  
05 FILLER PIC X(16) VALUE ALL SPACES.
```

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN ENTENDIDO

```
01 SUBCABECERA1.  
  05 FILLER      PIC X(6)  VALUE 'NOMBRE'.  
  05 FILLER      PIC X(9)  VALUE ALL SPACES.  
  05 FILLER      PIC X(8)  VALUE 'APELLIDO'.  
*  
01 SUBCABECERA2.  
  05 FILLER      PIC X(9)  VALUE ALL '-'.  
  05 FILLER      PIC X(3)  VALUE ALL SPACES.  
  05 FILLER      PIC X(15) VALUE ALL '-'.  
*  
01 DETALLE.  
  05 LT-NOMBRE   PIC X(9).  
  05 FILLER      PIC X(3)  VALUE ALL SPACES.  
  05 LT-APELLIDO PIC X(15).  
*  
01 SUBTOTAL.  
  05 FILLER      PIC X(6)  VALUE 'TOTAL '.  
  05 LT-NOMTOT  PIC X(9).  
  05 FILLER      PIC X(2)  VALUE ': '.  
  05 LT-NUMNOM   PIC 9(2).  
01 TOTALES.  
  05 FILLER      PIC X(15) VALUE 'TOTAL NOMBRES: '.  
  05 LT-TOTALES  PIC 9(2).  
*  
*****  
PROCEDURE DIVISION.  
*****  
* | 0000 - PRINCIPAL  
*-- |-----+-----><-----+-----*  
* | 1| EJECUTA EL INICIO DEL PROGRAMA  
* | 2| EJECUTA EL PROCESO DEL PROGRAMA  
* | 3| EJECUTA EL FINAL DEL PROGRAMA  
*****  
00000-PRINCIPAL.  
*  
      PERFORM 10000-INICIO  
*  
      PERFORM 20000-PROCESO  
          UNTIL FIN-ENTRADA  
*  
      PERFORM 30000-FINAL  
*  
*****  
* | 10000 - INICIO  
*-- |-----+-----><-----+-----*  
* | SE REALIZA EL TRATAMIENTO DE INICIO:  
* | 1| INICIALIZACIÓN DE ÁREAS DE TRABAJO  
* | 2| PRIMERA LECTURA DEL FICHERO DE ENTRADA  
* | 3| INFORMAMOS CABECERA Y ESCRIBIMOS CABECERA  
*****  
10000-INICIO.  
*  
      INITIALIZE DETALLE  
      WX-REGISTRO-ENTRADA  
  
      PERFORM 11000-ABRIR-FICHEROS  
      PERFORM LEER-ENTRADA  
  
      IF FIN-ENTRADA  
          DISPLAY 'FICHERO DE ENTRADA VACIO'  
  
          PERFORM 30000-FINAL  
      END-IF  
  
      MOVE WX-NOMBRE TO WX-NOMBRE-ANT
```

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN [ENTENDIDO](#)

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN ENTENDIDO

```
*****
*ESCRIBIR-CABECERAS.
*
      WRITE REG-LISTADO FROM CABECERA1 AFTER ADVANCING PAGE
      WRITE REG-LISTADO FROM CABECERA2
      WRITE REG-LISTADO FROM CABECERA3
      WRITE REG-LISTADO FROM SUBCABECERA1 AFTER ADVANCING 2 LINES
      WRITE REG-LISTADO FROM SUBCABECERA2

      MOVE 6          TO WC-LINEAS

*
*ESCRIBIR SUBTOTAL
*-- |-----+--><-----+-----*
* ESCRIBIMOS LINEA DE SUBTOTAL
*****ESCRIBIR-SUBTOTAL.
*
      MOVE WX-NOMBRE-ANT TO LT-NOMTOT
      MOVE WC-NOMBRES    TO LT-NUMNOM

      WRITE REG-LISTADO FROM SUBTOTAL AFTER ADVANCING 2 LINES

*
*21000 INFORMAR DETALLE
*-- |-----+--><-----+-----*
* INFORMAMOS LOS CAMPOS DE LA LINEA DE DETALLE CON LA
* INFORMACION DEL FICHERO DE ENTRADA
*****21000-INFORMAR-DETALLE.
*
      MOVE WX-NOMBRE    TO LT-NOMBRE
      MOVE WX-APELLIDO TO LT-APELLIDO

*
*ESCRIBIR DETALLE
*-- |-----+--><-----+-----*
* ESCRIBIMOS LA LINEA DE DETALLE
*****ESCRIBIR-DETALLE.
*
      WRITE REG-LISTADO FROM DETALLE

      IF NOT FS-LISTADO-OK
          DISPLAY 'ERROR AL ESCRIBIR DETALLE:'FS-LISTADO
      END-IF

      ADD 1          TO WC-LINEAS
      ADD 1          TO WC-NOMBRES
      ADD 1          TO WC-TOTALES

*
*LEER ENTRADA
*-- |-----+--><-----+-----*
* LEEMOS DEL FICHERO DE ENTRADA
*****LEER-ENTRADA.
*
      READ ENTRADA INTO WX-REGISTRO-ENTRADA

      EVALUATE TRUE
          WHEN FS-ENTRADA-OK
              CONTINUE
```

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN ENTENDIDO

```
END-EVALUATE

*
*****| 30000 - FINAL
*-- |-----+-----><-----+-----*
* | FINALIZA LA EJECUCION DEL PROGRAMA
*****| 30000-FINAL.
*

IF WC-LINEAS GREATER 60
    ADD 1      TO LT-NUMPAG

    PERFORM ESCRIBIR-CABECERAS
    PERFORM ESCRIBIR-SUBTOTAL
    PERFORM ESCRIBIR-TOTALES
ELSE
    PERFORM ESCRIBIR-SUBTOTAL
    PERFORM ESCRIBIR-TOTALES
ENDIF

PERFORM 31000-CERRAR-FICHEROS

STOP RUN

*
*****| ESCRIBIR TOTALES
*-- |-----+-----><-----+-----*
* | ESCRIBIMOS LA LINEA DE TOTALES DEL LISTADO
*****| ESCRIBIR-TOTALES.

MOVE WC-TOTALES      TO LT-TOTALES

WRITE REG-LISTADO FROM TOTALES

*
*****| 31000 - CERRAR FICHEROS
*-- |-----+-----><-----+-----*
* | CERRAMOS LOS FICHEROS DEL PROGRAMA
*****| 31000-CERRAR-FICHEROS.

CLOSE ENTRADA
LISTADO

IF NOT FS-ENTRADA-OK
    DISPLAY 'ERROR EN CLOSE DE ENTRADA:'FS-ENTRADA
ENDIF

IF NOT FS-LISTADO-OK
    DISPLAY 'ERROR EN CLOSE DE LISTADO:'FS-LISTADO
ENDIF
```

*

En el programa podemos ver las siguientes divisiones/secciones:
IDENTIFICATION DIVISION: existirá siempre.
ENVIRONMENT DIVISION: existirá siempre.
CONFIGURATION SECTION: existirá siempre.
INPUT-OUTPUT SECTION: en este ejemplo existirá porque utilizamos un fichero de entrada y uno de salida.
DATA DIVISION: existirá siempre.
FILE SECTION: en este ejemplo existirá pues utilizamos un fichero de entrada y uno de salida.

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN ENTENDIDO

En el programa podemos ver las siguientes sentencias:

PERFORM: llamada a párrafo
INITIALIZE: para inicializar variable
OPEN: "Abre" los ficheros del programa. Lo acompañaremos de "INPUT" para los ficheros de entrada y "OUTPUT" para los ficheros de salida.
DISPLAY: escribe el contenido del campo indicado en la SYSOUT del JCL.
MOVE/TO: movemos la información de un campo a otro.
PERFORM UNTIL: bucle
SET: Activa los niveles 88 de un campo tipo "switch".
READ: Lee cada registro del fichero de entrada. En el "INTO" le indicamos donde debe guardar la información.
EVALUATE TRUE: "Evalúa" si los niveles 88 por los que preguntamos en el "WHEN" están activados a "TRUE".
WRITE: Escribe la información indicada en el "FROM" en el fichero indicado.
STOP RUN: sentencia de finalización de ejecución.
CLOSE: "Cierra" los ficheros del programa.

Descripción del programa:

En el párrafo de **inicio**, inicializamos el registro de salida: WX-REGISTRO-SALIDA
Abriremos los ficheros del programa (OPEN INPUT para la entrada, y OUTPUT para la salida) y controlaremos el file-status. Si todo va bien el código del file-status valdrá '00'. Podéis ver la [lista de los file-status](#) más comunes.
Leemos el primer registro del fichero de entrada y controlaremos el file-status.
Además comprobamos que el fichero de entrada no venga vacío (en caso de que así sea, finalizamos la ejecución). Si todo ha ido OK guardaremos el nombre leído del fichero de entrada en WX-NOMBRE-ANT para controlar posteriormente el momento en que cambiamos de nombre.
Informaremos la parte genérica de la cabecera e inicializamos el contador de páginas.
Escribimos la cabecera por primera vez.

INFORMAR-CABECERA:

Recoge la fecha del sistema mediante un ACCEPT. La variable "DATE" es la fecha del sistema en formato 9(6) AAMMDD.
Para informar la fecha del listado formateamos la recibida del sistema a formato DD-MM-AA.

ESCRIBIR-CABECERAS :

Escribe las líneas de cabecera CABECERA1, CABECERA2, CABECERA3, SUBCABECERA1 y SUBCABECERA2.
En el WRITE de CABECERA1 vemos que utilizamos el "AFTER ADVANCING PAGE", esto significa que esta línea se escribirá en una página nueva. Lo podemos ver en el carácter ASA que aparece a la izquierda de esta línea en el fichero de salida que será un '1'.
En el WRITE de SUBCABECERA1 vemos que utilizamos "AFTER ADVANCING 2 LINES", esto significa que se escribirá una línea en blanco y después la línea de SUBCABECERA1. El carácter ASA que aparecerá será un '0'.
Informamos el contador de líneas a '6', pues la cabecera ocupa 6 líneas.

En el párrafo de **proceso**, que se repetirá hasta que se termine el fichero de entrada (FIN-ENTRADA), controlaremos:
Por un lado el número de líneas escritas, y en caso de superar un máximo (64 en nuestro caso) escribiremos otra vez las cabeceras en una página nueva (ver párrafo ESCRIBIR-CABECERAS).
Por otro lado la variable WX-NOMBRE, pues queremos escribir cada nombre en una página distinta. Cuando cambie el nombre escribiremos la línea de subtotales y volveremos a escribir cabeceras (que nos hará el salto de página).

Informaremos la línea de detalle y escribiremos en el fichero del listado.
Guardamos el último nombre escrito en WX-NOMBRE-ANT para controlar el cambio de nombre.
Leemos el siguiente registro del fichero de entrada.

Llamadas a párrafos:

ESCRIBIR-SUBTOTAL :

Informará el campo LT-NOMTOT con el nombre que hemos estado escribiendo, y LT-NUMNOM con el contador de registros escritos para ese nombre.
Escribirá SUBTOTALES dejando antes una línea en blanco (AFTER ADVANCING 2 LINES).

21000-INFORMAR-DETALLE :

Informamos los campos de la línea de detalle LT-NOMBRE y LT-APELLIDO con los campos

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN ENTENDIDO

En el párrafo de **FINAL**, finalizaremos el programa escribiendo la línea de subtotales que falta, y la línea de totales generales.

Controlaremos el número de líneas que llevamos escritas, pues para escribir la línea **SUBTALES** y **TOTALES** necesitamos 3 líneas.

En el párrafo de **ESCRIBIR-TOTALES** informaremos el campo **LT-TOTALES** con el contador de registros escritos en total.

Al inicio del artículo veíamos como quedaría el listado una vez "impreso", ya sea por pantalla o en papel. Vamos a ver como quedaría el fichero con los caracteres ASA que indican los saltos de linea.

Fichero de salida:

```
-----+---1---+---2---+---3---+---4---+
1 LISTADO DE EJEMPLO DEL CONSULTORIO COBOL
DIA: 22-03-11          PAGINA: 1
-----
ONOMBRE      APELLIDO
----- -----
ANA          LOPEZ
ANA          MARTINEZ
ANA          PEREZ
ANA          RODRIGUEZ
OTOTAL ANA : 04
1 LISTADO DE EJEMPLO DEL CONSULTORIO COBOL
DIA: 22-03-11          PAGINA: 2
-----
ONOMBRE      APELLIDO
----- -----
BEATRIZ     GARCIA
BEATRIZ     MOREDA
BEATRIZ     OTERO
OTOTAL BEATRIZ : 03
TOTAL NOMBRES: 07
```

Donde:

1 = salto de página.

0 = deja una línea en blanco justo antes.

- = deja dos líneas en blanco.

espacio = escribe sin dejar líneas en blanco.

Y listo! Si veis que no me he parado mucho en alguna cosa y queréis que explique más en detalle, dejad un comentario y lo vemos :)

Reacciones: Util (0) Interesante (1) Guay (1)

Publicado por Tallian



Etiquetas: ASA, COBOL, ejemplo, LISTADO, principiantes, Tallian

8 comentarios:



Anónimo dijo...

Muy buena página, nos está siendo muy útil el tutorial, sigue así! :D

18 de marzo de 2013, 13:08

Anónimo dijo...

Cacahuete

18 de marzo de 2013, 13:10

Anónimo dijo...

Gracias muy buena información, tengo una consulta qué significa TRK y CATLG en esta linea: DISP=(,CATLG),SPACE=(TRK,(50,10))
y no entendi muy bien el AFTER ADVANCING para que casos se usa?

13 de abril de 2013, 19:09

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

[MÁS INFORMACIÓN](#) [ENTENDIDO](#)

Tallian dijo...

Buenas Richard.

La primera sentencia lo que escribe en el fichero es la "marca" de salto de página.

La segunda una marca para que deje 2 líneas en blanco.

Son caracteres para la impresora, ya que en el fichero plano veremos todas las líneas seguidas.

Puedes verlo en el final del artículo, donde se ve como queda el fichero de salida y las marcas que aparecen con su significado.

Saludos!

27 de diciembre de 2013, 14:20

Juan Apastillado dijo...

HOLA SIGO ESTUDIANDO Y RECORDANDO LA PROGRAMACIÓN COBOL, TODO MUY BIEN, PERO SEGUN RECUERDO, TENGO QUE INDICAR EN LA "ENVIRONMENT DIVISION" QUE IMPRESORA VOY A OCUPAR, O SI LO VOY A MOSTRAR EN "PANTALLA, DISPLAY O COMO LE QUIERAN LLAMAR" SI ESTOY EN ERROR, POR FAVOR ACLARENMELO,
GRACIAS.

29 de junio de 2014, 1:27

Javier Cortijo López dijo...

Una cuestión, en el jcl, no faltaría el JOBLIB? El jcl no está accediendo a la librería donde está el programa objeto, es decir, ya compilado, hice un programa que vi aquí y me fallaba por eso, un saludo y si estoy en error, pues decírmelo sin problemas.

23 de agosto de 2017, 12:52

Tallian dijo...

Hola Javier.

Nosotros solemos incluir la JOBLIB en la cabecera del JCL para que aplique a todos los pasos. En los ejemplos no ponemos las cabeceras porque digamos que en función de donde trabajes pondrás unas opciones u otras. Por eso la echas en falta :-)

Pero no, no estás en ningún error. Más bien nosotros lo hemos dado por supuesto.

24 de agosto de 2017, 12:59



Introduce tu comentario...



Comentar como:

Mir2 (Google)

Cerrar sesión

Avisarme

[Entrada más reciente](#)

[Página principal](#)

[Entrada antigua](#)

[Suscribirse a: Envíar comentarios \(Atom\)](#)

ETIQUETAS

.NET 1-1-1 100 años 11 del 11 de 2011 2012 360 System 9 del 9 de 1999 AAAADD AAAAMDD abend Accenture ACCEPT Acciona Transmediterranea actualizaciones acuerdo tecnológico ADABAS ADD Adecco Alba Technology Alberto Larraz Alcalá de Henares alinear alinear alocar altair altamira Alvarito android android studio Ángel Martínez Navarro Anova IT AnyHelp App Apple Aragón arquitectura ASA ASCC asignatura asta AVG b37 Banco Espíritu Santo Banco Popular Banco Sabadell Banco Santander Bankinter Bankinter Barcelona BATCH BBK BBVA Besaide Bill Miller Biología BMC Software BMV BNDS Bolsa Mexicana de Valores BOOKMARKS borrar ficheros BPXWDYN Bull busquedas C/C++ cabecera Caixa Galicia CaixaBank Caja de Guadalajara Caja Sur Cejasol CALL canon

Este sitio emplea cookies de Google para prestar sus servicios, para personalizar anuncios y para analizar el tráfico. Google recibe información sobre tu uso de este sitio web. Si utilizas este sitio web, se sobreentiende que aceptas el uso de cookies.

MÁS INFORMACIÓN ENTENDIDO

espacios espera **estadísticas** estática EVALUATE EXEC CICS externalización facturación fecha DB2 fechas felicidad **Fichero** ficheros FILE file-status fill FILLER filtrar Fin FinancialTech FIND FINDREP Físicas FOR FORMATO FTP FUNCTION REVERSE fusión fusión servicios centrales futuro Gary Kasparov GDG generacional GET gnu cobol GOBACK grabar en fichero Grace Murray Hopper gratis gregoriana Heirloom Computing HEXLOC host HP HTML IBM IBM Websphere ICETOOL IDCAMS IDE IEBGENER IEFR14 iF IF THEN IFTHEN IMS INCLUDE index INDEXED BY indices infojobs informática informática móvil inner INPUT INREC INSERT INSPECT integración IS NUMERIC ISO 27001 ISRSUPC J2EE Java/J2EE JCL JES JES2 JFY JOB JOBLIB JOIN juliana justify katherine neville keys kmp Kubabank lecturas left Libros límite LINE Linux LISTADO LMMLIST lo mas leido LOAD **Loboc** longitud variable LOW-VALUES LPI LRECL Luis Krug Madrid **mainframe** Mainframe Transformation **manual** Mapfre Marcel Rafart Vicent market máscaras Master the mainframe matching Matemáticas MAX max-rc MAXCC máximos Medio Cudeyo mercado MERGE Micro Focus Micro Focus Test Server Microsoft migración Miguel Fito MIN MONTH mOSAIC move móviles msgclass MTC MTP MUL MySQL natural Nauta Capital **Noticias** notify novatos nube NUM STD COB ofertas OFFSET OMIT ONLINE opencobol operaciones ficheros optimización Optimyth Software oracle ordenar OUTFIL OUTLIM OUTREC outsourcing OVERLAY owner países paro parse **pepegan** perform personal communications PGM PHP PISTA. CILINDRO Pi-SQL plan de estudios play store plí precios prefix preguntas **principiantes** PRINT PROCEDURE procesadores especializados procesos por lotes PROCLIB programa programador purgar PUT QMF query Químicas QUIT RADIANS Rafael Achaerandio Randstad Rational Clear Case Raúl Gómez Martínez Razón READ READNEXT rearranque RECEIVE reclu-it record Red Hat REDEFINES reemplazar REGION rellena REMICS Renfe replace REPORT respuestas restart REWRITE REXX right RITSI Rod Adkins ROWSET RUN SOC7 Sap SCAN SCM SDSF SEARCH section secuencial **SELECT** SELECT MAX **semana** SEND SEQNUM SEQUENCE SER SGX SIN Singapur sistema sistemas sistemas informáticos SIVER SKIPREC sqlcode software inteligentes Sogeti SORT SPACE SPACES SPLICE SQL SQLCODE STARTBR STDDEV STEPLIB STOPAFT stream STRING STROBE Stuart McGill SUB subscripciones sueldo SUM SUM FIELDS Sun supercorte supercomputación SUSE SYSCOLUMNS SYSCOLUMNS SYSDUMP SYSEXEC SYSSIN SYSINDEXES sysname SYSOUT SYSPRINT SYSTABLES System Z sysuid Tallian TALLYING TAN tasas TASK teclado Tecalia THEN THRU TI TIC Ticjob TIER TIME PARM TIMESTAMP trabajo track **Trafuga** transacción Tres Cantos trigonometría TSO Tuxedo UAM UnixK union universidad unix UNLOAD UNNUM until UP BY UPDATE UTF-8 utilidades validar alfanumérico Vanson Bourne VARCHAR variables VARIANCE Vasco VASS VB vCOBOL visitas Visual Cobol R3 VMware VSAM VTL wait WEEK while Windows Windows Server x64 wiseri Workbench write XML XX YEAR z/Os zEnterprise Zurich

Contenido bajo licencia [Creative Commons](#).



www.consultoriocobol.com

Iconos dino de [Fast Icon Studio](#)

Con la tecnología de [Blogger](#).