

**lenguaje de control de
trabajos J.C.L.
OS/VS2**

JCL DEFINICIÓN	8
LENGUAJE	8
CICLO DE EJECUCIÓN DE UN PROCESO BATCH	8
SENTENCIAS	9
JOB	9
EXEC	9
DD	9
SENTENCIAS ESPECIALES	9
PROC	9
PEND	9
SENTENCIA DE COMANDOS	9
SENTENCIA NULA	9
SENTENCIA DELIMITADORA	9
SENTENCIA DE COMENTARIOS	9
CAMPOS	10
IDENTIFICADOR	10
NOMBRE LÓGICO	10
OPERACIÓN	10
PARÁMETROS	11
POSICIONALES	11
DE PALABRA CLAVE	11
SUBPARAMETROS (POSICIONALES Y DE PALABRA CLAVE)	12
COMENTARIOS	12
JOB	13
PARÁMETROS POSICIONALES	13
INFORMACIÓN DE CONTABILIDAD	13
NOMBRE DEL PROGRAMADOR	13
PARÁMETROS DE PALABRA CLAVE	14
MSGLEVEL	14
MSGCLASS	14
CLASS	14
NOTIFY	14
ADDRSPC	15
COND	15
DYNAMNBR	16
PRTY	16
TYPRUN	17
HOLD	17

SCAN	17
JCLHOLD	17
COPY	17
RD	17
RESTART	17
REGIÓN	18
TIME	18
PERFORM	18
SUBPARÁMETROS DE CONTROL DEL JES	19
ROUTE	19
JOBPARM	19
EXEC	20
PARÁMETROS POSICIONALES	20
PGM	20
PROC	20
PARÁMETROS DE PALABRA CLAVE	21
ACCT	21
ADDRSPC	21
COND	21
EVEN	21
ONLY	21
DPRTY	22
DYNAMNBR	22
PARM	23
PERFORM	23
RD	23
REGION	24
TIME	24
DD	25
CONCATENACIÓN DE DD.s	25
SENTENCIAS DD EN JOB	26
JOBLIB	26
JOBCAT	26
SYSCHK	26
SENTENCIAS DD EN EXEC	27
STEPLIB	27
STPCAT	27
SYSABEND	27
SYSUDUMP	27
PARÁMETROS POSICIONALES	28
*	28
DATA	28
DUMMY	29
DYNAM	29
PARÁMETROS DE PALABRA CLAVE	30

DDNAME	30
SYSOUT	30
COPIES	31
DEST	31
FREE	31
HOLD	31
UCS	32
OUTLIM	32
DSN	32
DSNAME	32
NOMBRE SIMPLE DE FICHERO	33
NOMBRE COMPUESTO DE FICHERO	33
NOMBRE DE FICHERO PARTICIONADO (PDS)	33
NOMBRE DE FICHERO TEMPORAL	33
NOMBRE DE FICHERO GENERACIONAL (GDG)	33
REFERENCIA A DD ANTERIOR	33
DISP	34
NEW	34
OLD	34
SHR	34
MOD	34
DELETE	34
KEEP	34
PASS	34
CATLG	34
UNCATLG	34
UNIT	36
SYSALLDA	36
TAPE	36
SYSALLDA	36
TAPE	36
TA80	36
TA90	36
SYSWK	36
WRK80	36
DEFER	36
AFINIDAD	36
AFF	36
SPACE	37
TRK	37
CYL	37
RLSE	37
CONTIG	37
MXIG	37
ALX	37
ROUND	37
ABSTR	37
LABEL	38
SL	38
SUL	38
AL	38
AUL	38
NSL	38
NL	38
LTM	38
BLP	38
PASSWORD	38
NOPWREAD	38
IN	38

OUT	38
EXPDT	38
RETPD	38
DCB	39
RECFM	39
U	39
V	39
VB	39
F	39
FB	39
FBA	39
FBM	39
LRECL	39
BLKSIZE	39
DSORG	39
PS	39
IS	39
DA	39
PO	39
DEN	40
PROCEDIMIENTOS	41
INCORPORADOS (IN-STREAM)	41
CATALOGADOS	41
MODIFICACIONES SOBRE UN PROCEDIMIENTO	42
MODIFICACIÓN O ADICIÓN DE UNA DD	42
CONCATENACIÓN DE UNA DD	42
PARÁMETROS SIMBÓLICOS	43
MODIFICACIÓN DE SENTENCIA EXEC	43
TERMINACIÓN DE UN TRABAJO	44
JCL ERROR	44
ENDED	45
ABENDED	45
CODIGOS ERRÓNEOS HABITUALES	46
RETORNO EN CALL AL IMS	46
SYSTEM CODES	49
USER CODES	51
UTILIDADES	53
ADUEMAIN	53
DBA00111	54
DBA00121	54
DBA00211	54
DBA00311	54

DBA00321	54
DFSRRRC00	55
DITTO	56
DSNUTILB	57
IBMABEND	58
IBMMENSA	58
IDCAMS	58
IEHLIST	58
IEBGENER	58
IEBCOPY	58
IEBCOMPR	58
IEFBR14	59
ISRSUPC	60
IKJEFT01	60
SISCOPY	60
SISGENER	61
SISPOPCL	61
SISPTOOL	61
PUTPARM	61
SORT	62
SORTLIB	62
SORTIN	62
SORTOUT	62
SORTWKnn	62
SYSIN	62
SORT	64
FORMAT	64
EQUALS	64
COPY	64
SENTENCIAS DE CONTROL(SORT)	65
ALTSEQ	65
INCLUDE	66
OMIT	67
INREC	68
OUTREC	69
MERGE	69
SUM	69
NONE	69
OPTION	70
COPY	70

NOEQUALS	70
EQUALS	70
NOLIST	70
LIST	70
SKIPREC	70
STOPAFT	70
VSAM	71
KSDS	71
ESDS	71
RRDS	71
LDS	71
INDICE CLUSTER	72
INDICE ALTERNATIVO (AIX)	72
PATH	72
IDCAMS	72
DEFINICIÓN DE VSAM	73
CLUSTER	73
DATA	73
INDEX	73
BORRADO DE VSAM	74
DELETE	74
CLUSTER	74
PURGE	74
ERASE	74
COPIADO DE VSAM	75
REPRO	75
INFILE	75
INDATASET	75
OUTFILE	75
OUTDATASET	75
FROMKEY	75
TOKEY	75
FROMADDRESS	75
TOADDRESS	75
FROMNUMBER	75
TONUMBER	75
COUNT	75
SKIP	75
IMPRESIÓN DE VSAM	76
PRINT	76
INFILE	76
INDATASET	76
CHAR	76
FROMKEY	76
TOKEY	76

FROMADDRESS	76
TOADDRESS	76
FROMNUMBER	76
TONUMBER	76
COUNT	76
SKIP	76
LISTADO DE CATALOGO DE VSAM	77
LISTCAT	77
LEVEL	77
ENTRIES	77
ALL	77
VOLUME	77
NAME	77
DEFINICIÓN DE UN AIX (ÍNDICE ALTERNATIVO)	78
DEFINE AIX	78
DATA	78
INDEX	78
CARGA DE UN AIX (INDICE ALTERNATIVO)	78
BLDINDEX	78
INDATASET	78
OUTDATASET	78

JCL DEFINICIÓN

LENGUAJE

JCL responde al acronimo de Lenguaje de Control de Trabajos (Job.s). Es la representación de mandatos en formato de sentencias que definen la forma en que deberán ejecutarse trabajos diferidos, también denominados BATCH.

Inicialmente el lenguaje se manifestaba por medio de fichas perforadas y por ello aun hoy en día existen expresiones referidas al lenguaje que provienen de aquella forma de trabajar, así pues a los trabajos diferidos, asociados a sentencias de control se les denomina trabajos por lotes, expresión que se remonta a los grupos de fichas perforadas que en un trabajo completo se denominaban lotes.

Se mantiene aun el formato de 80 posiciones como control de sentencias tal y como se estableció en su momento la anchura de fichas perforadas. También los hay que al referirse a una sentencia la mencionan como ficha de control.

Así pues conviene que nos familiaricemos con todas y cada una de estas expresiones al referirnos a JCL. JCL da información al sistema referente a:

- Información contable
- Nombre y numero de programas que deben ejecutarse
- El orden de ejecución de los mismos
- Ficheros o librerías que se precisan para el trabajo
- Soportes y periféricos necesarios
- Comentarios y mensajes al operador de consola
- Y otros

CICLO DE EJECUCIÓN DE UN PROCESO BATCH

La ejecución de trabajos Batch son realizadas en el sistema OS/VS2 bajo el subsistema JES2 (*Job Entry Subsystem*) o Subsistema de Entrada de Trabajos la forma en operar es como sigue:

ENTRADA	Se envía un trabajo al JES para su ejecución. Se identifica el trabajo con un nombre y numero permitiéndonos controlar en cualquier momento la ejecución del mismo. Las sentencias de JCL se introducen en un fichero SPOOL en cola de entrada .
CONVERSIÓN	Verificación sintáctica de las sentencias JOB. Revisión de la existencia de ficheros y librerías especificadas. Aportación de datos procedentes de procedimientos del sistema. Traspaso de los datos a la cola de ejecución de trabajos
EJECUCIÓN	Se carga el JCL generado en un iniciador. (Se entiende como iniciador a aquellas particiones que ejecutan un trabajo). Al abrir un iniciador se le especifican unas determinadas clases asociadas a unos subsistemas o elementos físicos de hardware y estos iniciadores son los encargados de recoger según orden de prioridades los trabajos y ejecutarlos.
SALIDA	Una vez efectuado el proceso de las sentencias, se da formato de salida al trabajo ejecutado y se procede al paso del mismo a colas de salida
IMPRESIÓN	Según la clase de salida que se especifique el las sentencias JOB se procederá a la impresión en una impresora determinada o visualización en un terminal o se guardara en cualquier medio que el sistema permita
BORRADO	El borrado suprime el control por parte del JES del trabajo realizado.

* El error producido en cualquier parte de las mencionadas, presupone la finalización anormal del trabajo conocida como JCL ERROR en el caso de error sintáctico o de validación de ficheros y de ABEND cuando el error proviene en tiempo de ejecución. La finalización correcta del trabajo es reconocida bajo la terminación ENDED.

SENTENCIAS

JOB

(TRABAJO) Identifica un trabajo . Es procesada por la parte de lectura del JES(*Reader*). Aporta un registro contable para el trabajo.

EXEC

(EJECUCIÓN) identifica un paso de trabajo a realizar. Se procesa por parte del JES en el apartado (*Converter*) destinado a modificar el JCL por rutinas y sentencias estándar del sistema . Se abre un registro contable que identifica el paso de trabajo.

DD

(DEFINICIÓN DE DATOS) Identifica un fichero al que acudiremos en el proceso de un programa. Como la sentencia anterior la ejecución de la sentencia es procesada en primer lugar por el apartado (*Converter*) del JES. En algunos casos si ello fuese preciso se acude al Gestor de Recursos del Sistema (SRM)

SENTENCIAS ESPECIALES

PROC

Determina el inicio de un procedimiento

PEND

Determina el final de un procedimiento

SENTENCIA DE COMANDOS

Se entiende como tal aquellas que permiten el envío de comandos al sistema.

SENTENCIA NULA

Es aquella que en las posiciones 1 y 2 tan solo disponen de forma doble del carácter / (//). quedando el resto de posiciones de la 3 a la 72 sin ningún contenido. Indica el final de un JOB

SENTENCIA DELIMITADORA

La sentencia delimitadora presupone el fin de datos introducidos en una sentencia DD con datos aportados dentro de las fichas de control. su formato es el de dos caracteres como (/*)

SENTENCIA DE COMENTARIOS

Es la que permite la incorporación de una ficha cuyo único contenido es el de un comentario quedando sin efectos ejecutables su contenido. se reconoce por sus primeras tres posiciones que quedan con el formato que se adjunta (/**)

CAMPOS

IDENTIFICADOR

El campo identificador se inicia en la posición 1 y existen tres tipos:

// **Sentencia JCL** (la mas comúnmente utilizada)

/* **Sentencia Delimitadora**

(Cierra la introducción de datos introducidos de forma directa a JCL)

//* **Comentario**

(El contenido no presupone mandato y podría ser suprimido sin alterar la ejecución del JCL)

NOMBRE LÓGICO

Identifica una sentencia para poder hacer referencia de la misma en cualquier situación.

Debe de preceder a los dos caracteres // de sentencia JCL (posición 3)

Pueden ser caracteres alfanuméricos y como especiales pueden utilizarse # @ \$

El primer carácter no debe ser numérico

Al campo nombre y como separación de la operación a la que hace referencia debe de precederle lo menos un blanco

OPERACIÓN

Tres son los tipos de operación validos como sentencia de control:

- JOB
- EXEC
- DD

Deben de ir a continuación del nombre y separados del mismo por uno o mas blancos

PARÁMETROS

Pueden ser de dos tipos:

POSICIONALES

El significado del parámetro viene dado por la posición que ocupa y la ausencia del mismo presupone el separarlo del parámetro siguiente por una (,) coma

Deben ir antes que los parámetros de palabra clave

La carencia del ultimo, o últimos parámetros posicionales sin continuación de parámetro posicional, o la falta de todos ellos no obliga la codificación de la (,) coma de separación

DE PALABRA CLAVE

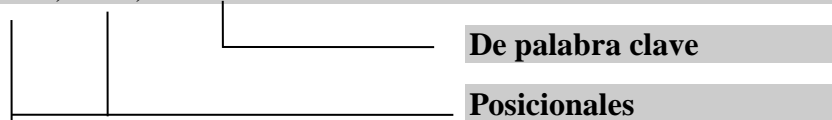
El significado lo determina el nombre de palabra clave y el valor que se le asocia

Deben de ir después de los parámetros posicionales si existen

Pueden codificarse en cualquier orden

No debe especificarse la ausencia de los mismos

//FEINA1 JOB A33-1,J.Ruiz,MSGCLASS=V



Los campos se separan por (,) comas

Deben iniciarse después del tipo de operación (JOB,EXEC,DD) separados de uno o mas blancos

No deben existir separación de blancos entre parámetros

Los parámetros no deben superar la posición 71

Caso de ser insuficiente las posiciones hasta la columna 71 podrá situarse una coma antes de la posición 72 y seguir codificando en la línea siguiente respetando las siguientes normas

No puede interrumpirse un parámetro y la interrupción de la línea será siempre al finalizar el mismo y después de la (,) coma

// en la primera y segunda posición

Podrá iniciarse la continuación a partir de la posición 4 y antes de la posición 16

Dependiendo de la instalación se asumen valores de algunos parámetros por defecto en cuyo caso no será necesario codificarlos de no ser que se precise variar su valor.

**//FEINA1 JOB A33-1,J.Ruiz,
// MSGCLASS=V**

SUBPARAMETROS (POSICIONALES Y DE PALABRA CLAVE)

Los parámetros de cualquier tipo pueden subdividirse en otros que se denominan subparámetros

Siguen las mismas normas que los parámetros

Deben de ir entre paréntesis salvo en aquellos casos que solo sea un subparámetro pudiendo prescindir de los mismos

//	DISP=OLD	}	posicionales
//	DISP=(NEW,CATLG,DELETE)		
//	DISP=(,CATLG)		
//	DCB=(LRECL=80,RECFM=FB)	}	de palabra clave (iguales)
//	DCB=(RECFM=FB,LRECL=80)		

COMENTARIOS

```
//FEINA1 JOB A33-1,J.Ruiz,MSGCLASS=V comentario
```

Si se codifican deben ir después del campo de parámetros y no pueden ponerse si estos no existen. Estarán separados por uno o mas blancos

Pueden codificarse hasta la posición 71

Aunque la columna 72 predispone a la continuación de un campo de comentario hay instalaciones que no lo contemplan por lo que se aconseja evitar codificar esa posición y utilizar la sentencia de comentario si fuese preciso ampliar el contenido del mensaje

A diferencia de la sentencia de comentarios el campo de comentarios precede a los parámetros y no debe codificarse sin la presencia de ellos y la sentencia de comentarios debe iniciarse en la primera posición con los identificadores (//*) siendo todo el contenido un comentario

JOB

Determina el principio de un trabajo y si no existen (//) el final de un trabajo anterior e inicio de otro.

Por cada trabajo solo debe existir una ficha JOB

Debe ser la primera ficha de codificación de un trabajo

Son partes de una sentencia JOB:

// en posición 1 y 2

Nombre lógico al que asociamos al trabajo a realizar

JOB

Parámetros posicionales y de palabra clave

Comentarios

Si se tienen que codificar apóstrofes (') o ampersands (&) como valor de los parámetros deberán codificarse dobles dando uno tan solo como resultado final

PARÁMETROS POSICIONALES

INFORMACIÓN DE CONTABILIDAD

Información contable que requiere la instalación (máxima longitud 142 caracteres)

NOMBRE DEL PROGRAMADOR

Persona responsable del trabajo a ejecutar (máxima longitud 20 posiciones)

```
//TRAB1 JOB DEPEXT,PUIG,.....  
//TRAB1 JOB DEPEXT,(PUIG,Planta 3a),.....
```

PARÁMETROS DE PALABRA CLAVE

MSGLEVEL

Indica si se han de imprimir o no las sentencias de JCL y los mensajes de asignación de recursos en el fichero de actividad del *log*. Es un parámetro compuesto de dos subparámetros posicionales el primero (Sentencias) acepta los valores :

- | | |
|---|---|
| 0 | Solo sentencias de Job |
| 1 | Sentencias de JCL y resolución de parámetros simbólicos por parte del JES |
| 2 | Sentencias de JCL |

El segundo (Mensajes):

- | | |
|---|--------------------------------------|
| 0 | Mensajes de JCL y todos si acaba mal |
| 1 | Todos siempre |

```
//TRAB1 JOB DEPEXT,PUIG,MSGLEVEL=(1,1).....
```

SI NO SE CODIFICA SE ASUMEN LOS VALORES POR DEFECTO DE LA INSTALACIÓN

MSGCLASS

Especifica la clase de salida al fichero de actividad de trabajo. Ha de ser cualquier carácter alfanumérico según decida la instalación

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,.....
```

CLASS

Especifica la clase de entrada y ejecución. La referida clase permite al Job situarse en cola del JES de trabajos . Ha de ser cualquier carácter alfanumérico según decida la instalación

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,.....
```

NOTIFY

Debe especificarse el usuario de TSO que deberá recibir aviso del final de la ejecución del trabajo. Máximo 7 caracteres definidos por la instalación como identificación de acceso al sistema (LOGON DE TSO)

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,.....
```

ADDRSPC

Especifica si el trabajo deberá ejecutarse en memoria REAL o VIRTUAL(Por defecto VIRTUAL)

ADDRSPC { VIRT }
 { REAL }

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// ADDRSPC=VIRT,.....
```

COND

Indica un código de condición para permitir o evitar ejecutar los diferentes pasos de un JOB en función del código de retorno que devuelve el sistema por la ejecución de cada uno de los pasos. EL JOB **DEJA DE EJECUTARSE** CUANDO EL CÓDIGO DE CONDICIÓN DEVUELTO CUMPLE EL CÓDIGO DEL PARÁMETRO COND UNA VEZ APLICADA LA OPERACIÓN DEL PARÁMETRO COND

Consta de dos subparámetros posicionales que pueden repetirse en diferentes comparaciones los dos subparámetros son :

código: Un numero de entre 0 y 4095 (los mas habituales son el (0, 4, 8, 12,16))

operación: Tipo de comparación

FINAL DE EJECUCIÓN CUANDO

Código devuelto	}	GT Mayor que	}	Código de parámetro COND
		GE Mayor o igual que		
		LT Menor que		
		LE Menor o igual que		
		EQ Igual a		
		NE No igual a		

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// COND=(9,GT),.....
```

El ejemplo cita que deberán ejecutarse los pasos posteriores si el actualmente en ejecución retorna un paso inferior a 9. Daría los mismos resultados la expresión que prosigue

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// COND=((9,EQ)(10,GT)),.....
```

SE RECOMIENDA EL USO DEL CÓDIGO DE CONDICIÓN A NIVEL DE PASO EN LUGAR DE JOB PUES ELLO PERMITE ADAPTARLO DE FORMA MAS PARTICULAR AL RESULTADO DE CADA EJECUCIÓN.

DYNAMNBR

Este parámetro está asociado con la asignación dinámica de ficheros y no se debe codificar en otras circunstancias

Asignación dinámica es aquella que efectuamos por medio del programa en ejecución que marca la pauta de que ficheros debe o no coger o crear.

Cada fichero a crear será una unidad de asignación al parámetro

Debe de asociarse el paso al parámetro para que el JOB sepa en que momento debe de reservar recursos para la asignación dinámica de ficheros.

DYNAMNBR.PASO01=10

El número máximo de ficheros a poder asignar de forma dinámica es de 1635

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// DYNAMNBR.PASO01=10,.....
```

LA ALOCACIÓN DINÁMICA SUELE ESPECIFICARSE EN LA SENTENCIA EXEC O POR MEDIO DE UNA DD (DYNAM)

PRTY

Determina la prioridad de ejecución del JOB en relación a otros en cola de ejecución.

El parámetro consta de dos subparámetros posicionales numéricos, el primero con valores comprendidos entre 0 y 14 y el segundo entre 0 y 15.

La prioridad final se obtiene del producto de $16 \times (\text{valor } 1) + (\text{valor } 2)$

```
//TRABA JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// DPTTY =(10,5),.....
```

SE RECOMIENDA NO UTILIZAR ESTE PARÁMETRO ,SINO ES POR CAUSAS JUSTIFICADAS Y ACEPTAR LA PRIORIDAD QUE ASIGNA EL JES 2 A RAÍZ DE LOS PARÁMETROS DE LA INSTALACIÓN PARA PRIORIDADES DE EJECUCIÓN SEGÚN LAS CLASES ASOCIADAS.

TYPRUN

Los posibles valores que adopta este parámetro:

HOLD

Deja el trabajo en HOLD (pendiente de ejecución para el JES) y queda a la espera que el operador de consola lo libere (Se hace habitualmente cuando el JOB requiere de unos recursos especiales que han de ser cedidos por Explotación.)

SCAN

Analiza sintácticamente el JOB Control.

JCLHOLD

El job queda retenido antes de ser procesado por el converter.No pasa a cola de selección hasta que el operador lo libera

COPY

Lista el JCL sin ejecutarlo

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// TYPRUN=SCAN,.....
```

RD

Especifica la posible forma de arrancar un JOB

$$RD = \left\{ \begin{array}{l} R \\ RNC \\ NC \\ NR \end{array} \right\}$$

R Rearranque automático combinado con la macro CHKPT
RNC Rearranque automático a nivel de paso (no de checkpoint)
NC No se permiten rearranques automáticos
NR No permite arranque automático, pero se puede tomar checkpoint para el arranque posterior

ESTE PARÁMETRO NO SUELE UTILIZARSE Y EN SU LUGAR SE UTILIZA FRECUENTEMENTE EL PARÁMETRO RESTART COMO REARRANQUE MANUAL.

RESTART

Especifica la posible forma de rearrancar un JOB de forma manual

$$RESTART = \left\{ \begin{array}{l} * \\ \text{nombre del paso} \\ \text{nombre del paso,nombre del paso dentro del procedimiento} \end{array} \right\} \text{ checkid}$$

- inicio desde el primer paso

Checkid especifica un checkpoin tomado anteriormente para el rearranque
debe de acompañarse con la **DD SYSCHK**

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// RESTART=PASPROC.PAS01
```

REGIÓN

En el caso de trabajar con memoria real **ADDRSPC=REAL** indica el total de memoria a utilizar

En el caso de trabajar con memoria virtual **ADDRSPC=VIRT** evita el exceso de paginación cuando se codifican valores bajos

La medida se expresa en valor superior a cero en numero par y como unidad K(1024 octetos)

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
// REGION=1024K
```

TAN SOLO DEBERÁ CODIFICARSE ESTE PARÁMETRO EN AQUELLOS CASOS QUE LA MEMORIA REQUERIDA EXCEDA LA ESTÁNDAR DE LA INSTALACIÓN

TIME

Especifica el tiempo máximo de CPU(tiempo en que el procesador utiliza recursos) autorizado para la ejecución del trabajo

TIME=([minutos]),[segundos])

ambos requieren valores numéricos el primero en rango comprendido entre 1 y 1439 y el segundo con valor entre 1 y 59 .Ejemplos TIME=1, TIME=(1,20), TIME=(,40)

Time=1440 (24 horas) presupone que no hay limite de CPU para ese JOB **(NO UTILIZAR)**

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,  
// TIME=(1,20)
```

LA AUSENCIA DE CODIFICACIÓN DE ESTE PARÁMETRO PRESUPONE EL TRABAJAR CON EL TIEMPO LIMITADO POR LA INSTALACIÓN

PERFORM

Grupo de rendimiento al que se aplica al JOB

PERFORM=numero

numero con valores comprendidos entre 1 y 255

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,  
// PERFORM=1
```

SE RECOMIENDA EL TRABAJAR CON LOS VALORES POR DEFECTO.

SUBPARÁMETROS DE CONTROL DEL JES

Deben de codificarse inmediatamente después de la ficha JOB. Damos detalle de algunas que son utilizadas de forma frecuente.

ROUTE

La ficha indica que la impresión debe encaminarse hacia una salida concreta

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C  
/*ROUTE PRINT R244
```

└─ La impresión debe encaminarse hacia la impresora Remota 244

JOBPARM

La ficha marca el valor de un parámetro del JES

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C  
/*JOBPARM LINECT=0
```

└─ No salta el papel por cada pagina nueva
LINECT= 88 predispone el salto de pagina pasadas 88 líneas

EXEC

Determina el inicio de un paso de trabajo y el programa o procedimiento catalogado a ejecutar .
El máximo de sentencias autorizado para un JOB es de 255

Las partes de una sentencia EXEC son:

// en las dos primeras posiciones
Nombre lógico del paso
Código de operación EXEC
Parámetros posicionales y de palabra clave
Comentarios

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214
//*
//PAS01 EXEC PGM=ACLP000.....
```

PARÁMETROS POSICIONALES

Se considera parámetro posicional de la sentencia EXEC la especificación del nombre del programa o procedimiento a ejecutar (todo y tratándose de un parámetro de palabra clave se considera posicional porque ha de ser el primero de los parámetros a especificar en la sentencia EXEC).

PGM

Se aplica cuando el paso deberá ejecutar un programa, asociando el nombre del programa a ejecutar.

Es un parámetro excluyente con el de PROC

PGM= { programa
*.nombre de paso.nombre DD

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214
//*
//PAS01 EXEC PGM=ACLP000.....
```

PROC

Se utiliza para hacer referencia a un procedimiento.

Es un parámetro excluyente con el de PGM

En este caso puede prescindirse de (PROC=) y especificar solo el nombre del procedimiento.

PGM= procedimiento
procedimiento

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
// NOTIFY=T515214,
//*
//PAS01 EXEC PROC=ACLPXXX.....
```

PARÁMETROS DE PALABRA CLAVE

A diferencia de los parámetros PROC y PGM que decíamos que había de ser uno de los dos el primero en codificarse en la ficha EXEC los restantes pueden codificarse en cualquier orden

Todos ellos son opcionales

ACCT

Da información contable del paso. (Tiene iguales características que el parámetro posicional contable de la sentencia JOB)

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//          NOTIFY=T515214  
//*  
//PAS01 EXEC PROC=ACLPXXX,ACCT=AP342,.....
```

ADDRSPC

Especifica si debemos ejecutar el paso en memoria real o virtual (Tiene características parecidas al mismo nombre de parámetro para la sentencia JOB)

COND

Determina cuando un paso deberá o no ejecutarse en relación al código devuelto por uno, varios o todos los pasos que anteceden al actual (Tiene características parecidas al mismo nombre de parámetro para la sentencia JOB)

Tiene algunas peculiaridades que no se aplican en la sentencia JOB y son:

$$\begin{aligned} \text{COND} &= \left\{ \begin{array}{l} (0,EQ) \\ (0,EQ,PAS01) \end{array} \right\} \\ \text{COND} &= \left\{ \begin{array}{l} \text{EVEN} \\ \text{ONLY} \end{array} \right\} \end{aligned}$$

EVEN

El paso se ha de ejecutar siempre, todo y que los anteriores acaben de forma incorrecta

ONLY

Solo se ejecuta si alguno de los pasos anteriores finalizase de manera incorrecta.

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//          NOTIFY=T515214  
//*  
//PAS01 EXEC PROC=AAAP000.....  
//PAS02 EXEC PROC=FINMALO,COND=(0,EQ,PAS01),.....  
//PAS03 EXEC PROC=FINMALO,COND=ONLY,.....  
//PAS04 EXEC PROC=SIFIN,COND=EVEN,.....
```

SI SE CODIFICO EL PARÁMETRO COND PARA LA SENTENCIA JOB QUEDARAN SIN EFECTO LOS PARÁMETROS COND DE LA FICHA EXEC

DPRTY

Determina la prioridad de ejecución del JOB en relación a otros en cola de ejecución.

El parámetro consta de dos subparámetros posicionales numéricos, el primero con valores comprendidos entre 0 y 14 y el segundo entre 0 y 15.

La prioridad final se obtiene del producto de 16x (valor 1)+(valor 2)

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214,  
//          DPRTY =(10,5),.....
```

SE RECOMIENDA NO UTILIZAR ESTE PARÁMETRO DE NO SER POR CAUSAS JUSTIFICADAS Y ACEPTAR LA PRIORIDAD QUE ASIGNA EL JES 2 PARA PRIORIDADES SEGÚN LAS CLASES DE EJECUCIÓN.

EL PARÁMETRO PRTY PARA LA FICHA JOB ES EL EQUIVALENTE A ESTE Y SU CODIFICACIÓN ANULA LA DE LA FICHA EXEC.

DYNAMNBR

Este parámetro esta asociado con la asignación dinámica de ficheros y no se debe codificar en otras circunstancias

Alocación dinámica es aquella que efectuamos por medio del programa en ejecución que marca la pauta de que ficheros debe o no coger o crear .

Cada fichero a crear será una unidad de asignación al parámetro

El numero máximo de ficheros a poder asignar de forma dinámica es de 1635

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214  
//*  
//PAS01 EXE PGM=AAAP000,DYNAMNBR=10,.....
```

LA ALOCACIÓN DINÁMICA SUELE ESPECIFICARSE EN LA SENTENCIA EXEC O POR MEDIO DE UNA DD (DYNAM)

ESTE PARÁMETRO TIENE EQUIVALENCIA AL DE IGUAL NOMBRE PARA LA FICHA JOB.(LA CODIFICACIÓN DEL PARÁMETRO EN JOB ANULA LA ESPECIFICADA EN EXEC)

PARM

Aporta información de datos externos solicitados por un programa en tiempo de ejecución

PARM=parametro.s

parámetros no puede exceder de un total de 100 caracteres

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214
//*
//PAS01 EXEC PGM=AAAP000,PARM='/ &CADENA'
```

CODIFICACIÓN EN PGM PLI DE RECEPCIÓN DE PARÁMETROS:

```
AAAP000: PROC (PARM) OPTIONS(MAIN)
DCL PARM CHAR (100) VAR;
DCL 1 WPARM      BASED(ADDR(PARM)),
    3 LONGIT     BIN FIXED(15),
    3 CADENA     CHAR(8);
```

PERFORM

Grupo de rendimiento al que se aplica al paso

PERFORM=numero

numero con valores comprendidos entre 1 y 255. Por defecto asume 1 para pasos de JOB no TSO y 2 para JOB.s de TSO

```
//TRAB1 JOB DEPEXT,PUIG,MSGCLASS=V,CLASS=C,
//          PERFORM=2
```

SE RECOMIENDA EL TRABAJAR CON LOS VALORES POR DEFECTO.

RD

Especifica la posible forma de arrancar un PASO

$$RD = \left\{ \begin{array}{l} R \\ RNC \\ NC \\ NR \end{array} \right\}$$

R	Rearranque automático combinado con la macro CHKPT
RNC	Rearranque automático a nivel de paso (no de checkpoint)
NC	No se permiten rearranques automáticos
NR	No permite arranque automático, pero se puede tomar checkpoint para el arranque posterior

ESTE PARÁMETRO NO SUELE UTILIZARSE Y EN SU LUGAR SE UTILIZA FRECUENTEMENTE EL PARÁMETRO RESTART COMO REARRANQUE MANUAL.

REGION

En el caso de trabajar con memoria real **ADDRSPC=REAL** indica el total de memoria a utilizar

En el caso de trabajar con memoria virtual **ADDRSPC=VIRT** evita el exceso de paginación cuando se codifican valores bajos

La medida se expresa en valor superior a cero en numero par y como unidad K(1024 octetos)

```
//TRAB1 JOB  DEPEXT,PUIG,MSGCLASS=V,CLASS=C,NOTIFY=T515214
//*
//PAS01 EXEC PGM=AAAP000,REGION=1024K
```

TAN SOLO DEBERÁ CODIFICARSE ESTE PARÁMETRO EN AQUELLOS CASOS QUE LA MEMORIA REQUERIDA EXCEDA LA ESTÁNDAR DE LA INSTALACIÓN

TIME

Especifica el tiempo máximo de CPU(tiempo en que el procesador utiliza recursos) autorizado para la ejecución del paso

TIME=([minutos]),[segundos])

ambos requieren valores numéricos el primero en rango comprendido entre 1y 1439 y el segundo con valor entre 1 y 59 .Ejemplos TIME=1, TIME=(1,20), TIME=(,40)

Time=1440 (24 horas) presupone que no hay limite de CPU para ese JOB
(NO UTILIZAR)

```
//TRAB1 JOB  DEPEXT,PUIG,MSGCLASS=V,CLASS=C
//*
//PAS01 EXEC PGM=AAAP000,TIME=(,15)
```

_____ 15 segundos

LA AUSENCIA DE CODIFICACIÓN DE ESTE PARÁMETRO PRESUPONE EL TRABAJAR CON EL TIEMPO LIMITADO POR LA INSTALACIÓN

DD

La sentencia DD responde al acronimo de (Definición de Datos). Nos permite definir los ficheros y librerías asociadas a un JOB o paso dentro del mismo

Es condición indispensable que cada fichero que tratemos deba estar representado por su ficha DD

No puede existir mas de una sola DD identificada con el mismo nombre lógico

La no especificación de nombre lógico en una ficha DD presupone la concatenación al fichero de la sentencia DD anterior

Las partes de una sentencia DD son:

// en las dos primeras posiciones
Nombre lógico de la DD
Código de operación DD
Parámetros posicionales y de palabra clave
Comentarios

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//          NOTIFY=T515214  
//*  
//PAS01 EXEC PGM=ACLP000.....  
//ENTRA DD DSN=..... FICHERO DE ENTRADA
```

CONCATENACIÓN DE DD.s

En ocasiones un paso puede precisar de mas de un fichero para una determinada entrada de datos y ello es posible por medio de la concatenación de DD.s. La forma en codificarse es:

```
//ESTEPLIB DD .....  
//          DD .....  
//          DD .....  
//XXXX DD.....
```

} concatenación de DD.s

En la concatenación de ficheros no es preciso que los mismos tengan la misma longitud ni el mismo factor de bloqueo pero en ambos casos habra que especificar en orden de mayor a menor

SENTENCIAS DD EN JOB

Las únicas DD asociadas a la ficha JOB son aquellas destinadas a definir librerías de acceso a las que deberán acudir los trabajos en tiempo de ejecución. El nombre lógico que las identifica es:

JOBLIB

La ejecución de un programa se inicia en la busca del objeto (código en lenguaje maquina) en las librerías estandars de la instalación (SYS1.LINKLIB) pero en según que casos puede sernos de utilidad el desplazar esa búsqueda a otras librerías.

La especificación de una o varias librerías no evita en ultimo caso el acudir a las estandars de la instalación si no se encontrase en ninguna de las referidas.

Ha de codificarse después de la ficha JOB y antes de cualquier paso EXEC

No puede utilizarse en procedimientos catalogados

La codificación de JOBLIB predispone a los pasos EXEC posteriores a que todos acudan a esas librerías para la obtención del objeto a ejecutar. Será excepción de lo dicho los pasos EXEC que dispongan de una DD STEPLIB en cuyo caso serán esas las librerías de captura.

JOBCAT

La diferencia de la JOBCAT con la JOBLIB radica en que mientras la anterior buscaba el objeto a ejecutar, esta marca el camino a seguir para la búsqueda y obtención del catalogo de ficheros. sigue las mismas pautas y en ultimo extremo acude a las estancadas de la instalación

Ha de codificarse después de la ficha JOB y de la JOBLIB y antes de cualquier paso EXEC

La codificación de JOBCAT predispone a los pasos EXEC posteriores a que todos acudan a esas librerías para la obtención del catalogo de ficheros. Será excepción de lo dicho los pasos EXEC que dispongan de una DD STEPCAT en cuyo caso serán esas las librerías de catalogo

SYSCHK

Define el fichero de grabación de CHECKPOINTS (puntos de control) de un programa que se guardan para re arranque posterior.

Debe ser anterior a cualquier paso EXEC de un JOB puesto que en re arranque y especificando la identificación del punto de control se deberá acudir a este fichero antes que al paso para obtener la información del programa que se pretende arrancar

SENTENCIAS DD EN EXEC

Al igual que las ficha JOB las EXEC tienen DD destinadas a definir librerías de acceso a las que deberán acudir los pasos de un trabajos en tiempo de ejecución. El nombre lógico que las identifica es:

STEPLIB

La ejecución de un programa se inicia en la busca del objeto (código en lenguaje maquina) en las librerías estandars de la instalación (SYS1.LINKLIB) pero en según que casos puede sernos de utilidad el desplazar esa búsqueda a otras librerías.

La especificación de una o varias librerías no evita en ultimo caso el acudir a las estandars de la instalación si no se encontrase en ninguna de las referidas.

Ha de codificarse después de la ficha EXEC aun que no tiene porque ser la primera DD

A diferencia de la JOBLIB puede utilizarse en procedimientos catalogados **(NO ES RECOMENDABLE Y EN SEGÚN QUE INSTALACIONES NO ESTA PERMITIDO)**

STEPCAT

La diferencia de la STEPCAT con la STEPLIB radica en que mientras la anterior buscaba el objeto a ejecutar, esta marca el camino a seguir para la búsqueda y obtención del catalogo de ficheros. sigue las mismas pautas y en ultimo extremo acude a las estandars de la instalación

Ha de codificarse después de la ficha EXEC aun que no tiene porque ser la primera DD.

La sentencia STEPCAT solo puede referirse a catálogos de usuario del tipo VSAM

SYSABEND

Determina el fichero donde el sistema efectuara el vuelco de memoria por terminación anormal
ABENDED

La información que aporta hace referencia a:

Núcleo del sistema

Área del programa problema

Tabla de Trace

SYSUDUMP

Determina el fichero donde el sistema efectuara el vuelco de memoria por terminación anormal
ABENDED

A diferencia de la anterior tan solo facilita información del Área del programa problema

PARÁMETROS POSICIONALES

Se considera parámetro posicional de la sentencia DD

*
DATA
DUMMY
DYNAM

*

Indica que los datos del fichero van a continuacion de la DD. La finalización de datos se indica con (/*) o con (//).

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//          NOTIFY=T515214  
/PAS01 EXEC PGM=ACLP000.....  
//SYSIN DD *  
SORT FIELDS=(1,1,CH,A)  
/*  
//ACCC DD
```

Delimitador

El parámetro y los datos que le preceden no pueden incorporarse dentro de un procedimiento pero si pueden referenciarse como modificación de los datos internos de un procedimiento.

```
:  
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//          NOTIFY=T515214  
//*  
//FERPROC PROC FICHA DE DEFINICION DE PROC  
//PAS001 EXEC SORT.....  
//          PEND FINAL DEL PROCEDIMIENTO  
//*  
//PAS01 EXEC FERPROC  
//PAS001.SYSIN DD *  
SORT FIELDS=(1,1,CH,A)  
/*
```

DATA

Este parámetro sigue las mismas normas que el anterior (*) tan solo a diferencia de ese no se consideran delimitadores de datos las (//)

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//          NOTIFY=T515214,  
//*  
/PAS01 EXEC PGM=ACLP000.....  
//SYSIN DD DATA  
SORT FIELDS=(1,1,CH,A)  
/*
```

Delimitador

DLM

Todo y siendo como es parámetro de palabra clave lo citamos entre los posicionales por estar ligado a (*) y DATA. Su codificación permite modificar los delimitadores de los dos citados (fin de entrada de datos (/*) o (/)) por otros dos diferentes caracteres. Ello puede ser de utilidad cuando entre los diferentes datos que se incorporen a nuestro trabajo existan los referidos /* y //, y ello provoque la finalización de datos antes de hora.

Si son caracteres especiales deberán codificarse entre apóstrofes

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214
/*
/PAS01 EXEC PGM=ACLP000.....
//SYSIN DD DATA,DLM='%%'
SORT FIELDS=(1,1,CH,A)
%%%
```

Delimitador modificado

DUMMY

El parámetro que traducido viene a ser equivalente a ficticio sirve para tratar como reales porque el programa los solicita con nombres lógicos ficheros que físicamente no existen o no existirán de salida (no se grabaran). De todas maneras el programa realizara el open y close de los ficheros como si los mismos existieran físicamente y sin que se produzca errores por no existir

Si la DD resulta ser de varios ficheros concatenados serán reales todos aquellos anteriores a la sentencia DUMMY y ficticios el codificado como Dummy y los posteriores

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214
/*
/PAS01 EXEC PGM=ACLP000.....
//SYSIN DD DUMMY
```

DYNAM

Hacíamos referencia a el parámetro DYNAMNBR en ficha JOB y EXEC cuando citábamos la asignación dinámica. Decíamos que el sistema debía reservar recursos para la posterior asignación dinámica y ello obligaba a realizar una estimación de cuantas asignaciones precisamos.

Deberán codificarse tantas DD DYNAM como posibles ficheros se puedan precisar

Es valida y equivale a la suma de ambas la combinación del parámetro DYNAMNBR de la ficha EXEC con el DYNAM de la ficha DD siendo el máximo numero de asignaciones autorizadas de 1635

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214
/PAS01 EXEC PGM=ACLP000.....
//DD1 DD DYNAM
//DD2 DD DYNAM
```

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214
/PAS01 EXEC PGM=ACLP000,DYNAMNBR=1 .....
//DD1 DD DYNAM
```

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
// NOTIFY=T515214,
/PAS01 EXEC PGM=ACLP000,DYNAMNBR=2.....
```

Los 3 preveen espacio para 2 ficheros

PARÁMETROS DE PALABRA CLAVE

Pueden codificarse en cualquier orden

Todos ellos son opcionales

DDNAME

Este parámetro sirve para diferir toda una DD hacia otro nombre lógico. Tiene utilidad cuando en un procedimiento queremos traspasar los datos de forma externa al procedimiento.

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//      NOTIFY=T515214,
//*
//FERPROC PROC                                FICHA DE DEFINICION DE PROC
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DDNAME=ENTR1      FICHERO DE ENTRADA 1
//      PEND                  FINAL DEL PROCEDIMIENTO
//PAS01 EXEC FERPROC
//PASO01.ENTR1 DD *
ALTAS111S 19970214
/*
```

Se procede a entrar los datos después del procedimiento haciendo referencia a la DD anterior

SYSOUT

Sirve para especificar las características de un fichero de salida que gestiona el JES. Consta de tres partes:

SYSOUT = ([CLASE] [,programa] [,form])

clase de salida
Programa que tratará la salida
Tipo de formulario de salida

Si se codifica **SYSOUT=*** se indica para este fichero la misma salida que la especificada en **MSGCLASS** de la ficha **JOB**

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//      NOTIFY=T515214,
//*
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD SYSOUT=*
//SYSPRINT DD SYSOUT=(A,,1224)
```

clase A y formulario 1224

clase de salida = V

COPIES

Sirve para indicar el numero de copias que queremos de salida. El valor máximo de copias de salida es de 255

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//          NOTIFY=T515214  
//*  
//PASO01 EXEC PGM=ACLP000.....  
//SYSPRINT DD SYSOUT=(A,,1224),COPIES=2
```

DEST

Da el destino de salida de un fichero del tipo SYSOUT

DEST= {


Rnnn	} Terminal remoto
RMnnn	
RMTnnn	
Unnn	Terminal local
LOCAL	Terminal por defecto de instalación
nombre	nombre dispositivo

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214  
//PASO01 EXEC PGM=ACLP000.....  
//SYSPRINT DD SYSOUT=(A,,1224),COPIES=2,DEST=R422
```

FREE

Libera un fichero y da acceso del mismo a otros usuarios que lo soliciten. Dos son sus posibles valores:

FREE= {


END	} 
CLOSE	

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214  
//PASO01 EXEC PGM=ACLP000.....  
//ENTRA1 DD DSN=AA1,FREE=CLOSE
```


HOLD

El fichero de salida queda retenido y no inicia su salida hasta que el operador de consola lo libera. Tan solo se puede utilizar en salidas de ficheros SYSOUT

HOLD= {

YES	} 
NO	

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214  
//PASO01 EXEC PGM=ACLP000.....  
//SYSPRINT DD SYSOUT=*,HOLD=YES
```

 Fichero de tipo SYSOUT

SE RECOMIENDA NO UTILIZAR SALVO EN CASOS JUSTIFICADOS ESTE PARÁMETRO Y DEJAR SU VALOR POR DEFECTO

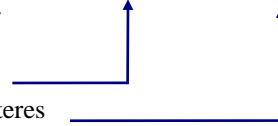
UCS

Indica el juego de caracteres que deberá utilizar una impresora

UCS= Juego de caracteres (,FOLD) ,VERIFY

Permite modificar el juego de caracteres por otro puede ser un ejemplo la transformación de caracteres mayúscula a minúscula

El operador debe verificar el juego de caracteres



```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//      NOTIFY=T515214  
//PASO01 EXEC PGM=ACLP000.....  
//SYSPRINT DD SYSOUT=*,UCS=(YN,,VERIFY)
```

SE RECOMIENDA NO UTILIZAR SALVO EN CASOS JUSTIFICADOS ESTE PARÁMETRO Y DEJAR SU VALOR POR DEFECTO

OUTLIM

Indica el numero máximo de registros que deben componer un fichero del tipo SYSOUT.
Cuando se sobrepasa este limite finaliza el JOB

OUTLIM= 1622

↑ valor máximo 16777215

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//      NOTIFY=T515214  
//PASO01 EXEC PGM=ACLP000.....  
//SYSPRINT DD SYSOUT=*,DEST=RMT266,OUTLIM=15000
```

LA NO CODIFICACIÓN DE ESTE PARÁMETRO PRESUPONE QUE NO HAY LIMITE DE SALIDA IMPRESA

DSN

DSNAME

(Data Set Name) Especifica el nombre físico del fichero

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,  
//      NOTIFY=T515214  
//PASO01 EXEC PGM=ACLP000.....  
//OUTPARM DD DSN=VCA.VCAD000.P010.&DATAP,DISP=(,PASS),  
//      SPACE=(00080,(000005,000015),,,),AVGREC=U,  
//      UNIT=(SYSALLDA,01),  
//      DATACLAS=,STORCLAS=,MGMTCLAS=,  
//      DCB=(LRECL=00080,RECFM=FB,BLKSIZE=0)
```

DSN=NULLFILE equivale a => DD DUMMY

NOMBRE SIMPLE DE FICHERO

Nombre de 8 posiciones alfanuméricas siendo el primer carácter alfabético o @,\$ó#

```
//OUTPARM DD DSN=ENTRA
```

NOMBRE COMPUESTO DE FICHERO

Consta de varios nombres simples separados por puntos

Cada nombre simple de 8 posiciones alfanuméricas siendo el primer carácter alfabético o @,\$ó#

El ancho total incluidos los puntos es de 44 posiciones

```
//OUTPARM DD DSN=ENTRA.V01.D7612
```

NOMBRE DE FICHERO PARTICIONADO (PDS)

Deberá especificarse el nombre del fichero particionado i el miembro de este entre paréntesis

Se rige por normas similares al SIMPLE y COMPUESTO

El nombre del miembro no puede exceder de 8 posiciones

El nombre del fichero particionado puede llegar incluso puntos a 44 posiciones mas paréntesis y 8 posiciones del miembro

```
//OUTPARM DD DSN=ENTRA.PLI(FITXER)
```

NOMBRE DE FICHERO TEMPORAL

Dos son las formas de referenciar un fichero temporal

Teclear el parámetro DSN y asignar el nombre del fichero temporal al que le antecede dos caracteres &&

Omitir el parámetro DSN(en este caso es el sistema el que asigna el nombre de fichero)

Puede especificarse un fichero temporal como particionado

El nombre del fichero no debe exceder de 8 sin contar los dos && (a parte paréntesis y nombre del miembro si se especifica como particionado)

```
//OUTPARM DD DSN=&&ENTRA
```

```
//OUTPARM DD DSN=&&ENTRA(ANTER)
```

NOMBRE DE FICHERO GENERACIONAL (GDG)

Fichero generacional es aquel que con un mismo nombre acepta un numero indeterminado de generaciones de manera que puede accederse a las diferentes versiones del mismo por el nombre y numero de generación o cabe especificar la ultima y anteriores de la siguiente forma:

- (0) Será la ultima de las generaciones
- (+1) para crear una nueva generación
- (-1) para la versión anterior a la actual
- (-2) para dos versiones anteriores

El numero máximo de generaciones es de 255. Por lo general se establece un numero inferior

```
//OUTPARM DD DSN=ENTRA(0)
```

REFERENCIA A DD ANTERIOR

Puede referenciarse un fichero utilizado anteriormente su codificación es como sigue

DSNAME=*.nompbrepaso.nombreDD

DSNAME=*.nombreDD (Solo cuando sea a una DD dentro del mismo paso)

```
//OUTPARM DD DSN=*.PAS1.INPUT
```

DISP

Especifica la disposición del fichero

Tiene un máximo de tres subparámetros posicionales:

El primero indica	DISPOSICIÓN EN ESTADO INICIAL
El segundo	DISPOSICIÓN POR TERMINACIÓN NORMAL
El tercero	DISPOSICIÓN POR TERMINACIÓN INCORRECTA

DISP= ({ NEW ,DELETE ,DELETE
OLD ,KEEP ,KEEP
SHR ,PASS ,CATLG
MOD ,CATLG ,UNCATLG
, ,UNCATLG })

NEW

El fichero es de nueva creación

OLD

El fichero existe y lo tomaremos para nuestro uso exclusivo

SHR

El fichero existe y lo tomaremos compartiendo el uso con otros

MOD

Si el fichero existe lo tomaremos para nuestro uso exclusivo. se posicionará en el ultimo registro y permite la grabación de nuevos registros tras los anteriores

Si el fichero no existe asume DISP=(NEW)

SI NO SE CODIFICA EL PRIMER PARÁMETRO (,xxx,xxx) SE ASUME **NEW**

DELETE

Al final del paso se elimina el fichero

KEEP

El fichero se guarda al final del paso . El sistema no guarda la dirección del fichero y por ello deberemos referenciar todos los datos para recuperar la información del mismo

PASS

Permite la utilización del fichero en pasos posteriores. En cintas y a diferencia de KEEP al finalizar el paso no se descarga la cinta y permite su uso posterior en otro paso.

CATLG

Los datos del fichero se mantienen y el sistema guarda información del lugar donde se guardan los datos para acceder a los mismos haciendo referencia tan solo al DSN del fichero.

UNCATLG

Se borra la dirección del catalogo del acceso a un fichero

Son valores asumidos por defecto:

La no codificación de DISP o DISP=(NEW)

DISP=(NEW,DELETE,DELETE)

DISP=(OLD)

DISP=(SHR)

DISP=(OLD,KEEP,KEEP)

DISP=(SHR,KEEP,KEEP)

```

//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//      NOTIFY=T515214
//*****
//**** SORT MOVIMIENTOS DE VENTAS/ANULACIONES POR OFICINA ****
//*****
//*-
//P005 EXEC PGM=IEFBR14
//DD1 DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,
//      DISP=(MOD,DELETE,DELETE)
//P010 EXEC PGM=SORT
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=(SHR)
//SYSOUT DD SYSOUT=*
//SYSIN DD DSN=EXPLOT.@BHE.CONTROL(VCA04201),DISP=(SHR)
//*-----
//*- SORT FIELDS=(56,03,BI,A,59,05,BI,A)
//*-----
//SORTWK01 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01)
//SORTWK02 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01)
//SORTWK03 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01)
//SORTWK04 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01)
//SORTWK05 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01)
//SORTWK06 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01)
//*----- MOVIMIENTOS DE VENTAS Y ANULACIONES DEL DIA
//SORTIN DD DSN=@BHE.VCA.VCA00007.&DATAP,DISP=(SHR)
//*----- VENTAS/ANULACIONES DEL DIA ORDENADO POR OFICINA
//SORTOUT DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,DISP=(,PASS),
//      SPACE=(01000,(015000,003000),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01),
//      DATACLAS=,STORCLAS=,MGMTCLAS=,
//      DCB=(RECFM=VB,LRECL=01000,BLKSIZE=0)
//ABEND010 EXEC PGM=IBMABEND,COND=(0,EQ,P010)
//*-----
//*- CATALOG DE LOS FICHEROS DEL PASO 10
//*-----
//P011 EXEC PGM=IEFBR14
//DD1 DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,
//      DISP=(OLD,CATLG),UNIT=(,DEFER)
//ABEND011 EXEC PGM=IBMABEND,COND=(0,EQ,P011)

```

Espera la finalización correcta del paso anterior y si es así CATALOGA el fichero

No especifica DISP asume valores por defecto (NEW,DELETE,DELETE)

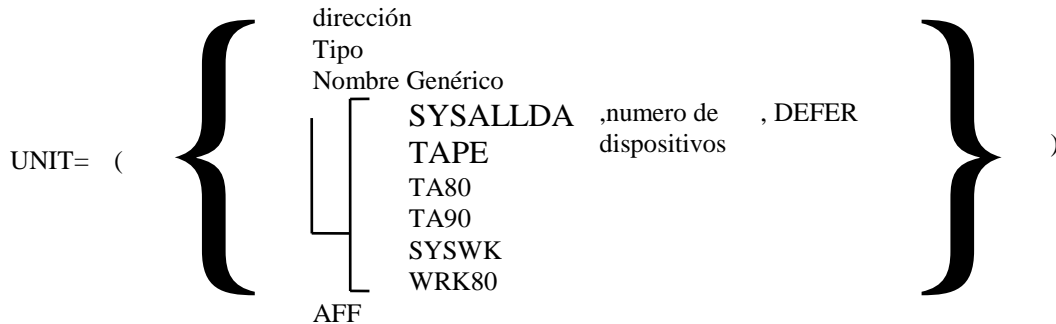
No especifica DSN el sistema asigna un nombre por defecto

Librerías del sistema que se recogen de lectura y en uso compartido con otros usuarios

Si el fichero existe lo borra/ Si no existe lo asume NEW y finaliza el paso de forma correcta

UNIT

Especifica tipo y numero de unidades que asignaremos al fichero. Son valores posibles de unidad a especificar:



dirección

Se indica una dirección determinada

Tipo

Nombre de sistema relacionado con IBM (2314,2400,3330,...)

Nombre Genérico Nombre que el sistema identifica con una serie de dispositivos como:

SYSALLDA

Dispositivo de acceso directo (Disco)

TAPE

Cinta

TA80

Cartucho.

TA90

Cartucho de alta densidad

SYSWK

Archivo de producción de tamaño pequeño

WRK80

Archivo de producción de tamaño mayor

Numero de dispositivos indica cuantos necesitamos para ese fichero (multivolumen)

DEFER

Pide que no se monte el volumen hasta la apertura del fichero

AFINIDAD

AFF

Solicita al sistema la asignación del mismo dispositivo que otro fichero especificado en una DD anterior (UNIT=AFF=DD1)

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//      NOTIFY=T515214
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,DISP=(,PASS),
//      SPACE=(01000,(015000,003000),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01),
//      DATACLAS=,STORCLAS=,MGMTCLAS=,
//      DCB=(RECFM=VB,LRECL=01000,BLKSIZE=0)
```

Especifica tipo y numero de unidades que asignaremos al fichero.Son valores posibles de unidad a especificar:

TRK
pistas

CYL
cilindros

longitud de bloque	Espacio en bloques (La unidad se asocia con el parámetro AVGREC)
cantidad primaria	Cantidad que se estima precisará el fichero
cantidad secundaria	Cantidad que se debe añadir en el caso de sobrepasar la cantidad primaria
directorio	Numero de bloques de 256 octetos que tendrá el directorio de un fichero particionado
índices	Numero de cilindros para un fichero indexado para ocupar al área de índices

RLSE
Solicita liberar el espacio no ocupado después de la creación del fichero

CONTIG
Pide que el espacio asignado para la petición primaria sea contiguo

MXIG
Solicita la asignación del área mayor de espacio libre (debe ser lo menos igual al área primaria)

ALX
Solicita la asignación del área mayor de espacio libre no inferior a 5 áreas contiguas de tamaño no menor al primario solicitado)

ROUND
 Tiene sentido en espacio por bloques y efectúa el redondeo a numero entero de cilindros.

ABSTR
Se pide la ubicación del fichero en una dirección específica dentro del volumen

dirección

Dirección de la primera pista

```
//TREBALL1 JOB PPMPP,'J.MIR',MSGCLASS=V,CLASS=C, NOTIFY=T515214
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,DISP=(,PASS),
//      SPACE=(01000,(015000,003000),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01),
//      DATACLAS=,STORCLAS=,MGMTCLAS=,
//      DCB=(RECFM=VB,LRECL=01000,BLKSIZE=0)
```

LABEL

Especifica tipo y numero de unidades que asignaremos al fichero. Son valores posibles de unidad a especificar:

LABEL= (secuencia ,SL ,PASSWORD ,IN ,EXPDT=aadd ,SUL ,NOPWREA ,OUT ,RETPD=dddd ,AL D ,AUL ,NSL ,NL ,LTM ,BLP)

secuencia Posición relativa en referencia al principio de la cinta. La información en cinta se ubica en bloques separados los unos de los otros por marcas de inicio y fin de fichero. este apartado determina si es el 1 (primer bloque),2 (segundo bloque)..... . Si no se especifica se asume el primero. Hay que tener en cuenta que si se especifican etiquetas estas presuponen un bloque mas a añadir en secuencia, por cada una de ellas..

SL	Fichero con etiquetas estandar de IBM(este se asume por defecto)
SUL	Fichero con etiquetas estándar de IBM y del usuario
AL	Fichero con etiquetas ANS
AUL	Fichero con etiquetas ANS y del usuario
NSL	El fichero no tiene etiquetas estándar
NL	El fichero no tiene ningún tipo de etiqueta
LTM	El sistema debe ignorar una marca de cinta de relleno en la cabecera de la cinta, si la encuentra en cintas sin etiquetas
BLP	En ficheros generados como BLP se pide no procesar las etiquetas, En los otros equivale a NL
PASSWORD	Especifica que no podrá accederse al fichero sin la palabra clave
NOPWREAD	No podrá accederse al fichero para actualizarlo pero si para leerlo
IN	Fichero de solo lectura
OUT	Fichero de solo escritura
EXPDT	Fecha en juliana de expiración del fichero (EXPDT=aaddd)
RETPD	Retención solicitada en días para un fichero (RETPD=dddd)

```
//P010 EXEC PGM=IDCAMS
//IN DD DSN=VCAA005,DISP=(OLD),
// UNIT=(TAPE,01),LABEL=(,NL),
// VOL=(SER=&VOL1),
// DCB=(BLKSIZE=&BLK,LRECL=&LON,RECFM=FB,DEN=3)
```

DCB

Descripción interna del fichero. Todos los subparámetro de este parámetro son de palabra clave

DCB= (RECFM = U ,LRECL=longitud ,BLKSIZE=l ,DSORG= PS ,DEN=densidad de
V longitud de la cinta
VB bloqueo
F
FB
FBA
FBC)

RECFM

U

Indefinido

V

Longitud variable

VB

longitud variable bloqueado

F

Longitud fija

FB

Longitud fija bloqueada

FBA

Longitud fija bloqueada con carácter de control de impresoras del tipo ASA

FBM

Longitud fija bloqueada con carácter de control de impresoras del tipo Maquina

LRECL

Longitud del registro

BLKSIZE

Longitud del bloqueo (en FBx múltiplo de LRECL)

DSORG

Tipo de organización del fichero

PS

Secuencial

IS

Secuencial indexado

DA

Organización directa

PO

Particionado

DEN

De aplicación para cintas identifica la densidad de grabación de las mismas.

Densidad = 0	Para cintas de 7 pistas de 200 bpi
Densidad = 1	Para cintas de 7 pistas de 556 bpi
Densidad = 2	Para cintas de 800 bpi
Densidad = 3	Para cintas de 9 pistas de 1600 bpi
Densidad = 4	Para cintas de 9 pistas de 6250 bpi

EL PARÁMETRO DCB TAN SOLO ES PRECISO EN LA GRABACIÓN DEL FICHERO , EN LA LECTURA DE CINTAS SIN ETIQUETAS Y EN AQUELLOS CASOS DE LECTURA QUE SE QUIERA DISPONER DE PARÁMETROS DIFERENTES A LOS QUE SE LE ASIGNARON EN LA GRABACIÓN.

```
//P010 EXEC PGM=IDCAMS
//IN DD DSN=VCAA005,DISP=(OLD),
// UNIT=(TAPE,01),LABEL=(,NL),
// VOL=(SER=&VOL1),
// DCB=(BLKSIZE=27900,LRECL=100,RECFM=FB,DEN=3)
```

El bloqueo optimo es el mas cercano a 27998 sin sobrepasarlo

PROCEDIMIENTOS

Definimos como procedimiento al grupo de sentencias de control dentro de uno o varios pasos que son llamados en tiempo de ejecución del JOB con un nombre único que identifica al procedimiento y se encarga el sistema de incorporar como pasos correspondientes al JOB en ejecución.

Restricciones a nivel de procedimientos:

- Un procedimiento no puede contener la llamada (EXEC) a otro procedimiento
- La sentencia JOB
- Los identificadores // (sentencia nula) ni /* (Delimitador)
- DD del tipo JOB (JOBLIB,JOBCAT....)
- Sentencias DD * (o su equivalente DD DATA)
- Sentencias de JES

Por sus características existen dos tipos de procedimientos:

INCORPORADOS (IN-STREAM)

Son los incorporados en un mismo fichero de ejecución del JOB y reclamados por la misma corriente de entrada. En primer lugar especificamos el procedimiento PROC y cuando finaliza este (Sentencia **PEND**), se crean los diferentes pasos que reclaman su ejecución. (El numero máximo de procedimientos dentro de un JOB es de 15)

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//          NOTIFY=T515214,
//FERPROC PROC                                FICHA DE DEFINICION DE PROC
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DSN=..... FICHERO DE ENTRADA
//PASO02 EXEC PGM=ACLP001.....
//ENTRA DD DSN=..... FICHERO DE ENTRADA
//          PEND                                FINAL DEL PROCEDIMIENTO
//          /*
//PAS01      EXEC FERPROC                        SOLICITA LA EJECUCION DEL
//          PROCEDIMIENTO (FERPROC) ANTERIOR
```

CATALOGADOS

Son los almacenados en librerías del sistema(librería de procedimientos) Su ejecución se reclama por medio de la sentencia EXEC que como parámetro posicional tiene el nombre de procedimiento ya sea PROC=NNNNN o bien NNNNN

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//          NOTIFY=T515214
//PASO01 EXEC FERPROC.....
           _____ = (PROC=FERPROC)
```

MODIFICACIONES SOBRE UN PROCEDIMIENTO

MODIFICACIÓN O ADICIÓN DE UNA DD

Los nuevos parámetros o modificación de los ya existentes presupone que deben de referirse una vez codificada la ficha EXEC que llama la ejecución del procedimiento.

Deberán especificarse primero las modificaciones de parámetros existentes y luego se citara la incorporación de nuevos parámetros.

Las parámetros a modificar deben expresarse en el mismo orden que en el procedimiento que deberá modificarse

La modificación de parámetros de una DD dentro de un paso deberá codificarse expresando primeramente el nombre del paso y con un (.) punto el nombre de la DD. Esta nomenclatura será idéntica para las DD a incorporar.

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//      NOTIFY=T515214
//FERPROC PROC                                FICHA DE DEFINICION DE PROC
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DSN=A1..... FICHERO DE ENTRADA 1 //PASO02
EXEC PGM=ACLP001.....
//ENTRA DD DSN=..... FICHERO DE ENTRADA
// PEND FINAL DEL PROCEDIMIENTO
//*
//PAS01 EXEC FERPROC
//PASO01.ENTRA DD DSN=B1
      └─┬─┘ Modifica el PASO01 la DD Entra para el fichero B1 en lugar del A1
```

CONCATENACIÓN DE UNA DD

La modificación de DD.s concatenada sigue una reglas:

- A) Para la modificación de la 1a. DD concatenada habrá que codificarse solo una DD de modificación
- B) Para las DD posteriores habrá que codificar una DD sin contenido por cada una de las que deban mantenerse hasta llegar a la DD que deba modificarse
- C) Si hubiese que sustituirse el valor a mas de una DD deberán aparecer en el mismo orden del procedimiento

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
//      NOTIFY=T515214
//FERPROC PROC                                FICHA DE DEFINICIÓN DE PROC
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DSN=A1..... FICHERO DE ENTRADA 1
//      DD DSN=A2..... FICHERO DE ENTRADA 2
//      PEND FINAL DEL PROCEDIMIENTO
//PAS01 EXEC FERPROC
//PASO01.ENTRA DD DSN=B1
```

A)Modifica el PASO01 la DD ENTRA para el fichero B1 en lugar de A1

```
//PASO01.ENTRA DD
//      DD DSN=B1
```

B) Mantiene el fichero A1 y sustituye A2 por B1

PARÁMETROS SIMBÓLICOS

Por lo general los procedimientos se crean para la utilización de los mismos por parte de diferentes trabajos y ello presupone la incorporación de parámetros variables según de donde proceda el trabajo a realizar.

Los parámetros se fijan de forma simbólica dentro del procedimiento para ser resueltos en tiempo de ejecución. Se especifican por medio del carácter (&) y antes de darles su valor definitivo se puede codificar su valor por defecto en la sentencia PROC.

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,NOTIFY=T515214
//FERPROC PROC U=SYSALLDA FICHA DE DEFINICION DE PROC
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DSN=A1,UNIT=&U..... FICHERO DE ENTRADA
// PEND FINAL DEL PROCEDIMIENTO
//PAS01 EXEC FERPROC
//PAS01 EXEC FERPROC,U=WRK80
```

sustitución del parámetro U por (WRK80)
acepta el valor asignado por defecto (SYSDA)

MODIFICACIÓN DE SENTENCIA EXEC

La modificación puede también efectuarse sobre parámetros de sentencia EXEC del propio procedimiento pasando nuevos o modificando parámetros existentes.

```
//TREBALL1 JOB PPPPP,'J.MIR',MSGCLASS=V,CLASS=C,
// NOTIFY=T515214
//FERPROC PROC, U=SYSALLDA FICHA DE DEFINICION DE PROC
//PASO01 EXEC PGM=ACLP000.....
//ENTRA DD DSN=A1,UNIT=&U..... FICHERO DE ENTRADA
// PEND FINAL DEL PROCEDIMIENTO
//PAS01 EXEC FERPROC,REGION=2048K
```

Asignamos mayor memoria

TERMINACIÓN DE UN TRABAJO

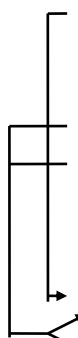
La finalización de un trabajo en ejecución en el JES puede tener un final diferente segun sean las circunstancias que motiven esa terminación .Vemos las tres posibles causas de terminación:

JCL ERROR

En validacion sintactica del JCL pueden existir errores por especificación de un parametro , por la existencia o no de un fichero o libreria especificada, o por no cumplir normas basicas de construcción de JCL estas circunstancias motivan la finalización con un mensaje asociado JCL ERROR, este mensaje a su vez se acompaña con la relación de mensajes que han sido los que han provocado la detección de sintaxis incorrecta de JCL

```
JES2 JOB LOG --- SYSTEM K158 ---NOD

14.21.40 JOB 9460   IEF4521 RUN6           JOB NOT RUN - JCL ERROR
14.21.40 JOB 9460   $HASP396 RUN6          TERMINATED
----- JES2 JOB STATICS -----
09 DEC 97 JOB EXECUTION DATE
    10 CARDS READ
    24 SYSOUT PRINT RECORDS
    0 SYSOUT PUNCH RECORDS
    1 SYSOUT SPOOL KBYTES
0.00 MINUTES EXECUTION TIME
1  //T5634GT JOB ALFPP,'DAU-P3 A.MARTI',MSGCLASS=V,CLASS=C,
   //      MSGLEVEL=(1,1),NOTIFY=T5634G
   /*ROUTE PRINT R244
2  //010          EXEC PGM=IEBGENER
3  //SYSPRINT      DD SYSOUT=*
4  //SYSOUT1       DD DSN=T5634.U12.V33,DISP=(SHR)
5  //SYSOUT2       DD DSN=T5634.U12.V33,DISP=(CATLG,DELETE),
                   SPACE=(TRK,(1,1),RLSE),UNIT=SYSALLDA,
6  //              DCB(FORMAT=FB,LRECL=80,BLKSIZE=4000)
7  //SYSIN         DD *
STMT NO. MESSAGES
2  IEF6471 FIRST CHARACTER OF NAME NOT ALPHABETIC OR NOT ...
5  IEF6321 FORMAT ERROR IN THE DISP FIELD
6  IEF6051 UNIDENTIFIED OPERATION FIELD
```



Son errores frecuentemente producidos al margen de los que son errores de tipo sintactico, aquellos que son resultantes de la duplicidad en la existencia de un fichero al intentar catalogarlo, o por lo contrario la no existencia de un fichero o libreria por su nombre incorrecto o por el borrado previo del mismo. Podemos localizar facilmente estos errores por el mensaje NOT FOUND o en el caso anterior por DUPLICATE NAME.

ENDED

La terminación correcta de un trabajo en ejecución en el JES2 se recibe por medio del mensaje ENDED

```
17.24.11 JOB17895 TSS7001I Count=00093 Mode=Fail Locktime=None Name=GERMAN
17.24.11 JOB17895 $HASP373 T5152C15 STARTED - INIT 52 - CLASS C - SYS SYST
17.24.14 JOB17895 -T5152C15 P001 P014 00 54 628 .00 .00
17.24.14 JOB17895 -T5152C15 P001 ABEND014 FLUSH 0 0 .00 .00
17.24.14 JOB17895 -T5152C15 ENDED. NAME=FFFD103 TOTAL TCB CPU TIM
17.24.14 JOB17895 $HASP395 T5152C15 ENDED
----- JES2 JOB STATISTICS -----
26 AUG 1997 JOB EXECUTION DATE
100 CARDS READ
542 SYSOUT PRINT RECORDS
0 SYSOUT PUNCH RECORDS
17 SYSOUT SPOOL KBYTES
0.05 MINUTES EXECUTION TIME
```

ABENDED

La ejecución incorrecta por errores detectados en el programa da como resultado el mensaje ABEND y junto al mensaje el código causante del error código que variará según la naturaleza del error. (En páginas posteriores hacemos relación de los códigos más comunes de retorno erróneo).

```
JES2 JOB LOG -- SYSTEM SYST -- NO

15.55.01 JOB14760 TSS7001I Count=00088 Mode=Fail Locktime=None Name=GERMAN
15.55.01 JOB14760 $HASP373 T5152C1M STARTED - INIT 37 - CLASS C - SYS SYST
15.55.05 JOB14760 IEA848I NO DUMP WAS PRODUCED FOR THIS ABEND, DUE TO
SYSTEM OR
15.55.05 JOB14760 IEF450I T5152C1M P014 P001 - ABEND=S722 U0000
REASON=00000000
15.55.05 JOB14760 -T5152C1M P001 P014 *S722 250 748 .00 .00
15.55.05 JOB14760 -T5152C1M P001 ABEND014 FLUSH 0 0 .00 .00
15.55.07 JOB14760 -T5152C1M ENDED. NAME=FFFD103 TOTAL TCB CPU TIM
15.55.07 JOB14760 $HASP395 T5152C1M ENDED
----- JES2 JOB STATISTICS -----
26 AUG 1997 JOB EXECUTION DATE
100 CARDS READ
20,248 SYSOUT PRINT RECORDS
0 SYSOUT PUNCH RECORDS
505 SYSOUT SPOOL KBYTES
0.10 MINUTES EXECUTION TIME
```

CODIGOS ERRÓNEOS HABITUALES

RETORNO EN CALL AL IMS

- AA • RESPUESTA PCB ALTERNATIVA NO PUEDE ESPECIFICAR CÓDIGO DE DESTINO
- AB • IOAREA NO ESPECIFICADA COMO UN PARÁMETRO DE LA -CALL-.
- AC • ERROR EN LA SECUENCIA JERÁRQUICA DE LAS -SSA-
 - EL NOMBRE DEL SEGMENTO EN LA SSA NO OCUPA 8 OCTETOS
 - EL NOMBRE DEL SEGMENTO EN LA SSA NO APARECE EN SENTENCIA SENSEG DE LA PCB
 - DOS SSA HACEN UNA COMPARACIÓN CON SEGMENTOS DEL MISMO NIVEL
 - UNA SSA UTILIZA UN SEGMENTO DE COMPARACIÓN DE UN NIVEL SUPERIOR AL DE LA SSA ANTERIOR
- AD • PARÁMETRO DE FUNCIÓN DE LA -CALL- INCORRECTO
 - LA FUNCIÓN DE LA CALL ESTA DEFINIDA EN UN CAMPO QUE NO TIENE 4 OCTETOS
- AF • FICHERO GSAM CON REG. DE LONG. VARIABLE INVALIDA
- AH • NO SE HA ESPECIFICADO LA -SSA- NECESARIA
- AI • ERROR DE APERTURA EN LA GESTIÓN DE DATOS
 - APERTURA INCORRECTA DE LA B.D. POR ESPECIFICACIONES INCORRECTAS EN LA SENTENCIA DE CONTROL O POR QUE NO EXISTE LA B.D.
- AJ • FORMATO DE CUALIFICACION DE LA -SSA- INCORRECTO
 - FALTAN LOS PARÉNTESIS O ESTÁN MAL COLOCADOS
 - EL NOMBRE DEL CAMPO NO TIENE 8 CARACTERES
 - OPERADOR INVALIDO
 - LA LONGITUD DEL CAMPO DE BÚSQUEDA ES MAYOR QUE LA DEFINIDA EN EL PARÁMETRO BYTES DE LA DBD
 - AL HACER UN INSERT SE USA UNA SSA CUALIFICADA
- AK • NOMBRE DE CAMPO INVALIDO EN LA -CALL-
 - EN LA CALL USAMOS UN CAMPO QUE NO ESTA EN LA FUNCIÓN FIELD DE LA DBD ASOCIADA AL SEGMENTO QUE ESTAMOS TRATANDO
- AL • UTILIZACIÓN INCORRECTA DE -PCB-TERMINAL- EN PROG.BATCH
 - SE ESTA UTILIZANDO UNA I/O PCB EN UN PROGRAMA BATCH (SE UTILIZA PTRPCB0). POSIBLEMENTE EL PROGRAMA ESTA COMPILADO SIN IMS
- AM • FUNCIÓN -CALL- INCOMPATIBLE CON OPCIÓN DE PROCESO
 - SE HA SOLICITADO EN LA CALL UNA OPCIÓN NO PERMITIDA POR EL PROCOPT DEL SEGMENTO TRATADO
- AO • ERROR DE ENTRADA-SALIDA, ISAM,OSAM,BSAM,VSAM
- AP • DEMASIADOS PARÁMETROS EN LA CALL DLI PARA TPPCB
- AQ • ERROR DE ENTRADA-SALIDA -READ-
- AR • ERROR DE ENTRADA-SALIDA -READ- CADENA MENSAJES O.K.
- AT • -IOAREA- DE USUARIO DEMASIADO LARGA
- AU • LONGITUD TOTAL DE LAS -SSA- ES EXCESIVA
- AY • TERMINAL LÓGICO DE LA -PCB- ASIGNADO A MAS DE 1 TERMINAL
- AZ • FUNCIÓN -PURGE- CON -PCB- INCORRECTA
- A1 • FUNCIÓN -CHANGE- CON NOMBRE TERMINAL LÓGICO DESCONOCIDO
- A2 • FUNCIÓN -CHANGE- CON -PCB- INCORRECTA
- A3 • FUNCIÓN -INSERT O PURGE- CON -PCB- SIN CONTENIDO
- A4 • SEGURIDAD DE DATOS. TERMINAL NO AUTORIZADO
- A5 • FUNCIÓN NI INSERT NI PURGE PARA EL PRIMER SEGMENTO DE SALIDA
- A6 • LONGITUD SEGMENTO SALIDA SUPERA EL LIMITE ESPECIFICADO
- A7 • NUMERO DE SEGMENTOS DE SALIDA INSERTADOS SUPERIOR AL LIMITE
- A8 • ISRT PCB ALTERNATIVA SEGUIDA ISRT IOPCB, O VICEVERSA
- A9 • PCB ALTERNATIVA HACE FALTA ESPECIFICAR (SAMETRM=YES)
- CA • COMANDO DESCONOCIDO

- CB • COMANDO NO AUTORIZADO PARA AOI
- CC • COMANDO EJECUTADO. HAY 1 O MAS RESPUESTAS
- CD • SECURITY VIOLATION
- CE • TRANSACCIÓN REPLANIFICADA DESPUÉS DE CMD
- CF • MENSAJE ANTERIOR A ULTIMO ARRANQUE DE IMS
- CG • TRANSACCIÓN GENERADA POR ÉXITO DE AOI
- CH • ERROR DE SISTEMA DETECTADO POR EL AOI
- CI • TRANSACCIÓN ANTERIOR A ULTIMA ARRANCADA DE IMS
- CJ • TRANSACCIÓN DE AOI REPLANIFICADA
- CK • TRANSACCIÓN DE AOI ANTERIOR A ULTIMA ARRANCADA DE IMS
- CL • TRANSACCIÓN DE AOI ANTERIOR A ULTIMA ARRANCADA,REPLANIF
- DA • EL CAMPO DE CLAVE DEL SEGMENTO HA SIDO CAMBIADA
 - EN UN DELETE O REPLAY SE INTENTA CAMBIAR EL CAMPO DE CLAVE POR EL QUE SE HABIA HECHO GHx
- DJ • NO HA EXISTIDO PREVIAMENTE UN -GET HOLD- CORRECTO
- DX • NO SE HAN SEGUIDO LAS REGLAS LÓGICAS DEL -DELETE-
- FA • OVERFLOW EN OPERACIÓN ARITMÉTICA EN UNA MSDB
- FC • CALL DE POSICIONAMIENTO SOLO PARA SEGMENTOS DIR. DEP
- FD • SE HA PRODUCIDO UN -DEAD-LOCK-
- FE • ERROR EN FSA: NO AFECTA A NOMBRE S DE CAMPOS
- FF • FALTA ESPACIO EN UNA MSDB
- FG • COMBINACIÓN DE CÓDIGOS -FE- I -FW-
- FH • BASE DE DATOS INACCESIBLE
- FI • ÁREA DE E/S NO ESTA EN LA REGIÓN DEL USUARIO
- FN • ERROR EN FSA: NOMBRE DE CAMPO INCORRECTO
- FP • DATOS HEXADECIMALES O DECIMALES INVÁLIDOS
- FR • SE HA SOBREPASADO EL NUMERO DE BUFFERS RESERVADOS
- FS • ÁREAS DEDB LLENAS
- FT • DEMASIADAS SSA EN UNA CALL
- FV • HA FALLADO UN VERIFY DE UNA MSDB
- FW • SE NECESITAN MAS RECURSOS DE LOS NORMALES
- GA • FRONTERA JERÁRQUICA CRUZADA HACIA UN NIVEL SUPERIOR
 - (ESTADO CORRECTO) SE HA HECHO UN GN SIN SSA Y EL SEGMENTO LEÍDO ES UNO QUE ESTA EN UN NIVEL SUPERIOR DEL QUE ESTABAMOS
- GB • FIN DE LA BASE DE DATOS
- GC • SE HA SOBREPASADO UNA FRONTERA DE UNIT OF WORK
- GD • FALTA SSA O SE HA PERDIDO LA POSICIÓN EN UN CALL
- GE • SEGMENTO NO ENCONTRADO
 - FINAL DE UN BUCLE CON GHN
- GG • POINTER INVALIDO EN UN SEGMENTO
- GK • OBTENIDO SEGMENTO DE TIPO DIFERENTE AL MISMO NIVEL
 - (ESTADO CORRECTO) MEDIANTE UN GN SIN SSA SE HA PASADO A UN SEGMENTO DIFERENTE PERO DE IGUAL NIVEL
- GL • CÓDIGO DE -LOG- INVALIDO COMO CODIGO DE USUARIO
- GP • NO SE HA ESTABLECIDO PREVIAMENTE EL SEGMENTO PADRE
 - LA PETICIÓN EXPRESADA EN EL GNP NO ES COMPATIBLE CON EL PADRE INDICADO
 - LA PRIMERA LECTURA EFECTUADA SE HACE CON UN GNP
 - SE HACE UN GNP Y EN LA GU ANTERIOR NO SE HABÍA ENCONTRADO EL SEGMENTO
- II • SEGMENTO DUPLICADO EN LA BASE DE DATO (ADICIÓN)
 - SE INTENTA INSERTAR UNA OCURRENCIA DE UN SEGMENTO CON UNA CLAVE QUE YA EXISTE Y DICHA CLAVE ESTA DEFINIDA COMO ÚNICA
- IX • NO SE HAN SEGUIDO LAS REGLAS LÓGICAS DEL -INSERT-
- LB • SEGMENTO DUPLICADO EN LA BASE DE DATOS (CARGA)
 - EN UN LOAD SE INTENTA CREAR UN REGISTRO CON CLAVE YA EXISTENTE Y

	ESTA HABÍA SIDO DEFINIDA
LC	• CONTENIDO DEL CAMPO CLAVE DEL SEGMENTO FUERA DE SECUENCIA
	• EN UN LOAD NO SE RESPETA LA SECUENCIA CRECIENTE
LD	• EL PADRE DE ESTE SEGMENTO NO HA SIDO CARGADO
	• EN UN LOAD SE OMITE EL PADRE DE UN SEGMENTO O NO SE RESPETA LA SECUENCIA JERÁRQUICA
LE	• SECUENCIA SEGMENTOS SENSITIVOS NO ES IGUAL EN LA -DBD-
	• EN UN LOAD SE INTENTA CREAR UNA OCURRENCIA DE UN SEGMENTO YA CREADO ANTERIORMENTE CUANDO YA HEMOS CREADO OTRO PARALELO
N	• CODIGO DE RETORNO INESPERADO AL ACTUALIZAR UN ÍNDICE
NE	• CALL DL.I- DE MANTENIMIENTO DE ÍNDICE NO ENCUENTRA SEGMENTO
NI	• MANTENIMIENTO DE ÍNDICE NO UTILIZABLE, O 2 SEGMENTOS 1 ÍNDICE
NO	• ERROR DE ENTRADA-SALIDA EN EL MANTENIMIENTO DE UN ÍNDICE
QC	• NO HAY MAS MENSAJES EN LA COLA DE ENTRADA (GET UNIQUE)
QD	• NO HAY MAS SEGMENTOS PARA ESTE MENSAJE (GET NEXT)
QE	• FUNCIÓN -GN- UTILIZADA ANTES DE ESTABLECER POSICIÓN CON -GU-
QF	• LONGITUD DEL SEGMENTO DE MENSAJES ES MENOR DE 5 CHARS
QH	• TERMINAL LÓGICO O CÓDIGO TRANSACCIÓN DESCONOCIDO POR EL IMS-VS
RX	• NO SE HAN SEGUIDO LAS REGLAS LÓGICAS DEL -REPLACE-.
UC	• REGISTRO DE -CHECKPOINT- GRABADO AL -UCF JOURNAL-
UR	• -RESTART- BAJO -UCF- DEL PROGRAMA DE CARGA INICIAL
US	• -STOP- PROGRAMA DE CARGA INICIAL
UX	• -CHECKPOINT AND STOP-
V1	• LONGITUD INVALIDA PARA SEGMENTO DE LONGITUD VARIABLE
XA	• INTENTO DE PASAR -SPA- DESPUÉS DE RESPUESTA DEL TERMINAL
XB	• PROGRAMA TRATA DE RESPONDER AL TERMINAL.-SSA- PASADA
XC	• CAMPO -Z1- DEL MENSAJE, RESERVADO -IMS- CON CONTENIDO
XD	• PROCESO DE -CHECKPOINT FREEZE O DUMPQ-
XE	• INTENTO DE INSERTAR SPA A UNA PCB ALTERNATIVA (EXPRESS=YES)
XF	• INSERT SPA A UNA PCB ALTERNATIVA SIN (ALTRESP=YES)
XG	• TRANSACCIÓN CON LONGITUD INCORRECTA O VARIABLE DE -SPA-
XX	• ERROR INTERNO DEL GSAM
X1	• ERROR DE ENTRADA-SALIDA DEL IMS-VS CON EL -SPA-
X2	• PRIMER -INSERT- AL CÓDIGO TRANSACCIÓN PCB NO ES UN -SPA-
X3	• -SPA- INVALIDA. 6 PRIMEROS -BYTES- MODIFICADOS
X4	• INSERT- CÓDIGO TRANSACCIÓN PCB, NO CONVERSACIONAL CON SPA
X5	• VARIAS -SPA- INSERTADAS PARA UN CÓDIGO TRANSACCIÓN -PCB-
X6	• NOMBRE DE CÓDIGO TRANSACCIÓN INCORRECTO, INSERTADO EN LA SPA
X7	• LONGITUD DE LA -SPA- INCORRECTA. PRIMEROS 6 -BYTES-
X8	• ERROR COLA DEL SISTEMA. -SPA-, CÓDIGO TRANSACCIÓN -PCB-.
X9	• INCOMPATIBILIDAD PROGRAMA CONVERSACIONAL I -CALL PATH-

SYSTEM CODES

	SA0A-10	<ul style="list-style-type: none"> • PROBLEMA DE BLKSIZE EN LA DCB
IEC151I	SA13	<ul style="list-style-type: none"> • ERROR POSICIONAMIENTO CINTA. ERROR HARDWARE CINTA.
IEC030I	SB37	<ul style="list-style-type: none"> • FALTA ESPACIO EN CREACION ARCHIVO DE SALIDA.
IEC031I	SD37	<ul style="list-style-type: none"> • NO HAY PISTAS ALTERNATIVAS EN CREACION ARCHIVO.
IEC030I	SE37	<ul style="list-style-type: none"> • FALTA ESPACIO EN DISCO AL CREAR EL ARCHIVO DE SALIDA
	S001	<ul style="list-style-type: none"> • CINTA ETIQUETA NO SE CORRESPONDE • SI ES DISCO S001-4 ARCHIVO VACIO
	S002	<ul style="list-style-type: none"> • PUEDE SER QUE LA DCB TENGA ERRONEO EL LRECL O EL BLKSIZE (EJ.- SISGENER, ARCH. SALIDA DISTINTOS ATRIBUTOS RESPECTO A ENTRADA).
	S0CB	<ul style="list-style-type: none"> • ERROR DE EASY.
	S0C4	<ul style="list-style-type: none"> • LIMITE DE TABLA SOBREPASADO (PROTECTION EXCEPTION). • FICHERO DE SALIDA DE TAMAÑO INSUFICIENTE
	S0C4-04	<ul style="list-style-type: none"> • LA B.D. O EL AREA ESTEN PARADAS • PTR'S EN BMP MAL DIRECCIONADOS. • ACCESO PLITDLI-SISMDLI SIN PUNTERO B.D. • SISGENER CON DCB I001 DIFERENTE DE 0001.
	S0C7	<ul style="list-style-type: none"> • SE HA PRODUCIDO UNA SITUACION DE 'DATA EXCEPTION'
IEC141I	S013	<ul style="list-style-type: none"> • INTENTO LECT. ARCH. VB CON INDICACION FB (O VICEV.)
IEC141I	S013-14	<ul style="list-style-type: none"> • DIRECTORIO DE LA LIBRERIA LLENO
IEC141I	S013-18	<ul style="list-style-type: none"> • DATA SET NOT FOUND O MIEMBRO DE LA LIBRERIA.
IEC141I	S013-20	<ul style="list-style-type: none"> • POSIBLEMENTE, BLKSIZE NO MULTIPLO DE LRECL
IEC141I	S013-34	<ul style="list-style-type: none"> • DATA SET VACÍO
IEC141I	S013-4C	<ul style="list-style-type: none"> • SORT.- HAY UN PROBLEMA DE BLKSIZE. PONER MAYOR BLKSIZE AL SORTIN01
	S106	<ul style="list-style-type: none"> • POSIBLEMENTE NO PUEDE EJECUTAR UN PROGRAMA O MODULO, DEBIDO A QUE LA LIBRERÍA OBJETO ESTA "COGIDA" POR UN PROCESO ESPECIAL (POR EJ.: SE ESTA COMPRIMIENDO)
	S18A	<ul style="list-style-type: none"> • POSIBLEMENTE ERROR DE LONGITUD DEL BLKSIZE CON DD'S, CONCATENADAS. LA LONGITUD DE LA PRIMERA DD, HA DE SER LA MAS GRANDE DE TODAS.
IEC143I	S213-04	<ul style="list-style-type: none"> • DATA SET NOT FOUND
IEC210I	S214-10	<ul style="list-style-type: none"> • 'TRANSMIT' ARCHIVO DE SALIDA A CINTA
IEC022I	S137-0C	<ul style="list-style-type: none"> • ERROR AL GRABAR LA ETIQUETA EN LA CINTA
	S222	<ul style="list-style-type: none"> • JOB CANCELADO POR EL OPERADOR
IEC023I	S237-04	<ul style="list-style-type: none"> • POSIBLE ERROR DE HARDWARE. UN BLOQUE SE HA PERDIDO • NO SE CORRESPONDE EL TOTAL DE ETIQUETAS DE COLA (VOLVER A EJECUTARLO)
	S322	<ul style="list-style-type: none"> • PARÁMETRO 'TIME' DE FICHAS JOB O EXEC INSUFICIENTE. • POSIBLE 'LOOP' DENTRO DEL PROGRAMA
IEC145I	S413-0C	<ul style="list-style-type: none"> • ERROR DE HARDWARE EN LA LECTURA DE UN LABEL DE CINTA.
IEC146I	S513-04	<ul style="list-style-type: none"> • SON DOS ARCHIVOS QUE ESTÁN EN LA MISMA CINTA. HACER UN SISGENER DE UNO SOBRE OTRO SOPORTE.
IEC147I	S613-04	<ul style="list-style-type: none"> • PROBABLE ERROR CINTA IRRECUPERABLE. PEDIR OTRA CINTA.
	S613-10	<ul style="list-style-type: none"> • ERROR EN LA ESCRITURA DE UN LABEL EN CINTA
IEC026I	S637	<ul style="list-style-type: none"> • DD CONCATENADA CON ARCHIVOS EN SOPORTES DIFERENTES, O CON DCB'S DISTINTOS.
IEC215I	S714	<ul style="list-style-type: none"> • ERROR EN LA ESCRITURA DE UN TM EN CINTA.
	S722	<ul style="list-style-type: none"> • LIMITE DE LÍNEAS DE SALIDA ESPECIFICADA PER OUTLIM EXCEDIDA. • SI BMP, PUEDE FALTAR PARAM. REGION (1024 -> 2048) • SI BMP, SE HA PODIDO COMPILAR COMO NO-IMS.
IEA700I	S80A	<ul style="list-style-type: none"> • FALTA MEMORIA VIRTUAL PARA EJECUTAR EL PROGRAMA AMPLIAR PARÁMETRO REGIÓN DE 512K EN 512K

IEA703I	S806-04	<ul style="list-style-type: none"> PROGRAMA O MODULO DE ACCESO NO ENCONTRADO EN LA LIBRERÍA ESPECIFICADA. VOLVER A ARRANCAR, NO SE HA PODIDO ACCEDER AL MODULO (DEVICE END MISSING) PUEDE SER QUE FALTE LA STEPLIB <-----
IEC149I	S813-04	<ul style="list-style-type: none"> NO SE CORRESPONDE LA ETIQUETA CON EL NOMBRE DEL ARCHIVO, O BIEN CINTA DEFECTUOSA.
IEC028I	S837-08	<ul style="list-style-type: none"> EL ARCHIVO DE SALIDA OCUPA MAS DE 5 VOLÚMENES, PONER VOL=(,,99)
	S913	<ul style="list-style-type: none"> POSIBLEMENTE INCOMPATIBILIDAD CON TOP-SECRET INTENTO DE ACCESO A ARCHIVO NO PERMITIDO

USER CODES

	U0002	• CAÍDA DE LA REGIÓN DE CONTROL DEL IMS
ICE015A	U0015	<ul style="list-style-type: none"> • VARIABLE RECORD TOO SHORT • EN UN SORT O MERGE ERROR FÍSICO DE CINTA. HABRÁ QUE PONER (OPTION VLSHRT) Y ELLO PROVOCA QUE EL PROCESO CONTINUÉ IGNORANDO EL PROBLEMA. SI CON ESTA OPCIÓN DIESE UN U0068 CONFIRMA QUE ES ERROR DE CINTA, PROBAR CON LA COPIA DE SEGURIDAD SI LA HAY.
	U0043	• DATA SET ATTRIBUTES.- DCB SALIDA MAS PEQUEÑA QUE LA DE ENTRADA
	U0046	• “SORT CAPACITY EXCEEDED” AÑADIR SORTWK
	U0053	• NO HA HABIDO PREVIAMENTE UN -GET HOLD- CORRECTO NO RESERVA EL REGISTRO PARA MODIFICARLO
	U0061	<ul style="list-style-type: none"> • SI ES UN PROBLEMA DE I/O EN SORTIN PUEDE SER DEBIDO A LA DCB O LOS DATOS. • ARCHIVO SORTIN VACIO.
	U0068	• ERROR EN MERGE: UNO DE LOS ‘SORTIN0X’ NO ESTA CLASIFICADO POR LOS TÉRMINOS MARCADOS EN EL SYSIN DEL MERGE. (OJO, PUEDE QUEJARSE DEL SORTIN01 Y ESTAR MAL CLASIFICADO EL SORTIN03 (POR EJEMPLO)).
	U0073	<ul style="list-style-type: none"> • BTS => HACER LOGOFF I LOGON • DLI-BATCH => FALTA PARÁMETRO DBRC=N EN FICHA EXEC
	U0100	• RESTART INCORRECTO (REPASAR)
	U0102	• CHKPT NO ENCONTRADO
	U0114	• DENTRO DE UN SORT: PARÁMETROS EN ORDEN ERRÓNEO (POR EJEMPLO, EN UNA ‘INCLUDE COND’).
	U0134	• SE ESTA PARANDO EL LA REGIÓN DE IMS DE EJECUCIÓN (PUEDE DAR, POR EJEMPLO, EN BTS Y PARANDO IMS).
	U0144	• BMP CANCELADO POR HABER DEMASIADAS REGIONES ABIERTAS
	U0242	• EL PARÁMETRO DIRCA ES DEMASIADO PEQUEÑO PARA LA PSB. AUMENTAR EL DIRCA
	U0260	• EL NUMERO DE PARÁMETROS EN LA LLAMADA AL IMS ES DEMASIADO GRANDE O DEMASIADO PEQUEÑA
	U0261	• PARÁMETROS PASADOS POR PARM AL BMP ERRÓNEOS
	U0271	• I/O ERROR EN UNA OPERACION DE CHECKPOINT
	U0428	• NOMBRE DE PSB NO DEFINIDA
	U0430	• FALTA MEMORIA. PONER REGIÓN O AUMENTAR SU VALOR.
	U0452	• HAY UN PROGRAMA BMP PRIORITARIO QUE PROVOCA LA CANCELACIÓN. LOS DOS DEBEN UTILIZAR LA MISMA PSB.
	U0456	• PSB PARADA.
	U0457	• SE HA ARRANCADO UNA PSB QUE JA ESTABA ACTIVA EN OTRA REGIÓN DEL IMS.
	U0458	• BASE DE DATOS PARADA
	U0474	• SE HA PARADO LA REGIÓN EN QUE SE EJECUTABA PGM
	U0476	• NO COINCIDEN LOS PUNTEROS DE B.D. DE LA PSB, CON LOS ESPECIFICADOS EN EL PARM DEL PROGRAMA.
	U0630	<ul style="list-style-type: none"> • POSIBLE FALTA DEL PARAM. IMS=EXP EN PROC DLIBATCH. • POSIBLE MEZCLA PARAM. IMS PRUEBAS Y REAL.
	U0688	<ul style="list-style-type: none"> • PROGRAMA DE CONTROL NO ACTIVO. • POSIBLEMENTE EL IMS ESTE PARADO. • ESTAS EJECUTANDO CON CLASE QUE NO ES DE ESTE IMS
	U0757	• CAIDA DEL IMS POR PROBLEMAS EN LAS COLAS
	U0775	• PROBLEMAS DE ESPACIO EN MEMORIA (NO SE TOMA CHECKP)
	U0777	• LA REGIÓN DONDE SE PROCESA EL PROGRAMA HA SIDO CANCELADA PARA EVITAR UN ‘DEAD LOCK’(BESO DE LA

	MUERTE)
U0778	<ul style="list-style-type: none"> • HA HABIDO UN ERROR AL HACER BACKOUT DEL BMP POR ERROR DEL PROGRAMA O DEL SISTEMA
U0844	<ul style="list-style-type: none"> • FALTA DE ESPACIO EN B.D.
U0850	<ul style="list-style-type: none"> • EN BMP'S Y MPP'S: B.D. ERRONEA (MAL INICIALIZADA, REDEFINIDA CON VERSIÓN DISTINTA EN ACBLIB, ETC).
U0929	<ul style="list-style-type: none"> • LA PSB O LA DBD NO SE ENCUENTRAN EN LAS LIBRERÍAS CORRESPONDIENTES.
U0999	<ul style="list-style-type: none"> • DETECTADO CÓDIGO DE RETORNO INCORRECTO
U1006	<ul style="list-style-type: none"> • EL NUMERO DE BUFFERS SOLICITADOS EXCEDE LA CANTIDAD DE LOS DISPONIBLES. ES POSIBLE QUE HAYAN DEMASIADOS BMP.s EN EJECUCIÓN
U1008	<ul style="list-style-type: none"> • FALTA DE ESPACIO EN BUFFERS (NBA-OBA) AMPLIAR ESPACIO. • EN BMP CONTRA FAST PATH FALTA CHECKPOINT DE FINAL-OK PER A LLIBERAR BUFFERS.
U1033	<ul style="list-style-type: none"> • SI SE PRODUCE EN ON-LINE, PUEDE SER DEBIDO A QUE LA TRANSACCION NO ESTE EN LA CLASE QUE LE CORRESPONDA.
U3001	<ul style="list-style-type: none"> • TERMINACION ANORMAL POR SIGNAL ERROR
U3042	<ul style="list-style-type: none"> • SE NOTO AL EJECUTAR UN PROGRAMA ON-LINE IMS-DB2 EN REGIONES DE CLASE 6 CUANDO LA CLASE DE SU REGIÓN DEBERÍA SER 9. • EL TAMAÑO DE LA REGIÓN ES PEQUEÑO • HAY PROBLEMAS CON LAS STEPLIBS • OJO CON LAS LIBRERÍAS LOAD
U3303	<ul style="list-style-type: none"> • HAY UNA BASE DE DATOS PARADA.
U3501	<ul style="list-style-type: none"> • LONGITUD DEL FRONT-END INCORRECTA. • POSIBLEMENTE, FALTA ACTIVAR EL BIT EOM DEL MENSAJE DE SALIDA.
U4000	<ul style="list-style-type: none"> • ERROR IMPREVISTO EN MODULO ERRORES DE PL/I. (POR EJ. FALTAN (SOBRAN) PARAM. PASO AL MODULO) • SI LO DEVUELVE UN PGM. BATCH, PUEDE PASARSE EL PARÁMETRO: PARM='/NOSTAE' ==> DA UNA MEJOR INFORMACIÓN DE LO OCURRIDO. • PGM BATCH Y ACOMPAÑADO DE 'B37', REVISAR SPACE ARCHIVOS DE SALIDA (FALTA DE ESPACIO). • TAMBIEN: EN CALL AL DLI (IMSWDLI), EL 3ER. PARAM. (IOAREA-BD) ESTE DECLARADO DE LONGITUD VARIABLE (EJ.- DCL IO_BD CHAR(1500) VAR; .. DARA PROBLEMAS).
U4036	<ul style="list-style-type: none"> • MIRAR COMO ESTÁN COMPLETADAS LAS DEFINICIONES DE LAS SSA'S: (POR EJEMPLO, EL NOMBRE DE SEGMENTO HA DE OCUPAR 8 POSICIONES I SI NO SE HA DE COMPLETAR CON BLANCOS)

UTILIDADES

Las utilidades son un conjunto de programas codificado y comprobados que realizan el conjunto de funciones útiles.

ADUEMAIN

Utilidad que permite la ejecución de comandos DB2 y la obtención de resultados en la salida de la ddname SYSREC

```
//P010 EXEC PGM=ADUUMAIN,PARM='&IDDB2,VCAD010,NEW,,MSGLEVEL(1)'
/*-
//SYSPRINT DD SYSOUT=*
//UTPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=P,HOLD=YES,FCB=6666
//SORTWK01 DD SPACE=(27906,(000042,000126),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK02 DD SPACE=(27906,(000042,000126),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK03 DD SPACE=(27906,(000042,000126),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SYSREC DD DSN=@BHE.VCA.VCAD010.P010.&DATAP,DISP=(,PASS),
// SPACE=(02000,(000054,000162),RLSE,,),AVGREC=U,
// UNIT=(),
// DATACLAS=,STORCLAS=,MGMTCLAS=,
// DCB=
//SYSCNTL DD DUMMY,DSN=,
// DCB=(RECFM=VB,LRECL=00200,BLKSIZE=0)
//SYSIN DD *
UNLOAD
SELECT *
FROM DBT1.ACAT01_CARDSN
ORDER BY KACAT01
/*
```

DBA00111

Procedimiento que realiza una reorganización e IMAGE COPY posterior de un Tablespace no particionado DB2. Los parámetros se corresponde en igual nombre y posición a los referidos en la utilidad **DBA00321**

DBA00121

Procedimiento que realiza una reorganización e IMAGE COPY posterior de un Tablespace particionado DB2. Los parámetros se corresponde en igual nombre y posición a los referidos en la utilidad **DBA00321**

DBA00211

Procedimiento que realiza una actualización de las estadísticas de catalogo (**RUNSTATS**) DB2 referente a su TABLESPACE, TABLA e INDICES. Los parámetros se corresponde en igual nombre y posición a los referidos en la utilidad **DBA00321**(No se especifica UNIT por no hacerse IMAGE COPY).

Es conveniente la actualización del catalogo después de una carga de datos cuando estos se consideran estables para optimizar los caminos de acceso a los datos.

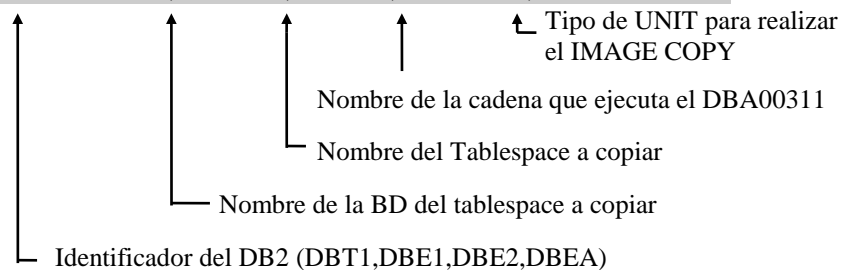
DBA00311

Procedimiento que realiza IMAGE COPY de un Tablespace no particionado DB2. Los parámetros se corresponde en igual nombre y posición a los referidos en la utilidad **DBA00321**

DBA00321

Procedimiento que realiza IMAGE COPY de un Tablespace particionado DB2

```
//P010 EXEC DBA00321, DB2ID=&DB2,BD=&BD,TS=&TS,UID=&UID,UNIT=&UNIT
```

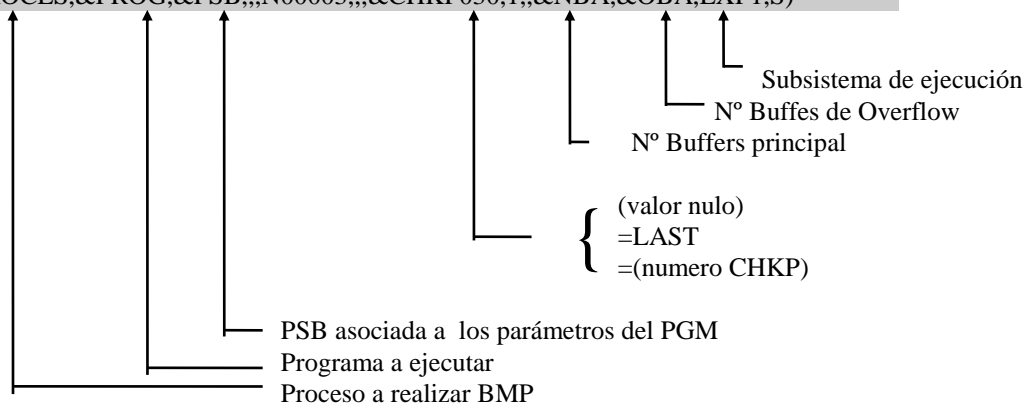


DFSRR00

Este programa de utilidad sirve para la ejecución de programas en el entorno IMS

```
//P030 EXEC PGM=DFSRR00,
//      PARM=(BMP,TAAP605,TAAP605,,,N00003,,,&CHKP030,1,,&NBA,&OBA,EXP1,S)
//STEPLIB DD DSN=IMSEXP1.RESLIB,DISP=(SHR)
//*      DD DSN=DB2EXP1.DSNLOAD,DISP=(SHR)
//*      DD DSN=DB2EXP1.DSNEXIT,DISP=(SHR)
//IMS    DD DSN=IMSEXP.PSBLIB,DISP=(SHR)
//      DD DSN=IMSEXP.DBDLIB,DISP=(SHR)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=00133,BLKSIZE=1330)
//PLIDUMP DD SYSOUT=P,HOLD=YES,FCB=6666
//SYSUDUMP DD SYSOUT=P,HOLD=YES,FCB=6666
//ZMENSAJ DD SYSOUT=*,DCB=(LRECL=133,RECFM=FBA,BLKSIZE=1330)
//TAAA011 DD DSN=ACA.ACAD100.TAUDEL.&DATAP,DISP=(OLD)
//TAAA012 DD DSN=ACA.ACAD100.DELTAU.&DATAP,DISP=(,CATLG,DELETE),
//      SPACE=(00265,(000282,000846),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01),
//      DATACLAS=,STORCLAS=,MGMTCLAS=,
//      DCB=(RECFM=FB,LRECL=00265,BLKSIZE=0)
```

```
//      PARM=(&PROCES,&PROG,&PSB,,,N00003,,,&CHKP030,1,,&NBA,&OBA,EXP1,S)
```



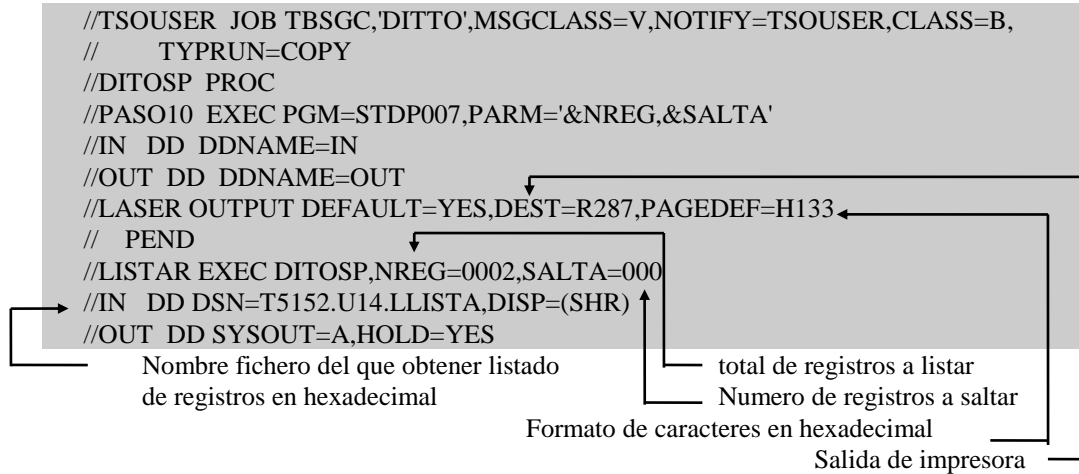
NBA

OBA La utilización optima de los buffers pasa por utilizar los asignados como buffers de principal NBA sin utilizar los dedicados al Overflow OBA .Se aconseja el uso de NBA=25 i OBA=15 y como maximo NBA=30 y OBA =15

Los buffers OBA no deben utilizarse normalmente por ser recursos compartidos por todos los BMPs en ejecución. Un BMP accede a buffers de Overflow en el momento que detecta el código de estado FW y hasta el momento en que se toma nuevo CHEKPOINT.

DITTO

Se obtiene un listado de los registro de un fichero en formato hexadecimal. Pueden saltarse registro y acotar la totalidad de los mismos por medio de los parámetros que se especifican.



DSNUTILB

Utilidad que permite la ejecución de comandos DB2 y la obtención de resultados en la salida de la ddname SYSREC

```
//P010 EXEC PGM=DSNUTILB,REGION=&SIZE,PARM='&SYSTEM,&UID1,&UTPROC'
//SYSPRINT DD SYSOUT=*
//UTPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=P,HOLD=YES,FCB=6666
//* PASO 1 : EJECUTAR EL COMANDO LOAD.
//SORTWK01 DD SPACE=(16384,(000200,000600),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK02 DD SPACE=(16384,(000200,000600),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK03 DD SPACE=(16384,(000200,000600),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK04 DD SPACE=(16384,(000200,000600),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SYSREC DD DSN=&PRF.AC134545.&DATAP,DISP=(SHR)
//SYSUT1 DD DSN=&PRF.ACID871.SYSUT1.&DATAP,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(000036,000108),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTOUT DD DSN=&PRF.ACID871.SORTOUT.&DATAP,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(000004,000012),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SYSERR DD DSN=&PRF.ACID871.SYSERR.&DATAP,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(000200,000600),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SYSMAP DD DSN=&PRF.ACID871.SYSMAP.&DATAP,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(000200,000600),,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SYSIN DD *
LOAD DATA INDDN SYSREC RESUME YES LOG NO
INTO TABLE DBE2.ACIT02
( T02EMPR POSITION(1) DECIMAL
, T02CENT POSITION(3) DECIMAL
, T02CALE POSITION(6) CHAR (7)
, T02FVAL POSITION(13) DECIMAL
, T02ITOT POSITION(18) DECIMAL
, T02NTOT POSITION(26) DECIMAL
, T02IPAR POSITION(31) DECIMAL
)
/*
/* EL LOAD PUEDE DAR COND.CODE 4 (ES CORRECTO)
//ABEND010 EXEC PGM=IBMAABEND,COND=(4,GE,P010)
/*-----
/* IMAGE COPY DE @TABLESPACES DEL DBE2
/*-----
//P101 EXEC DBA00311,DB2ID=DBE2,BD=ACIB01,TS=ACIE01S,UID=ACID8711,
```

DSN SYSTEM(DBE2)
RUN
PROGRAM(DSNTEP2) -

LIB('DB2EXP2.RUNLIB.LO
AD')

DELETE FROM
DBE2.ACIT02
COMMIT;

IBMABEND

```
//ABEND010 EXEC PGM=IBMABEND,COND=(0,EQ,P010)
```

IBMMENSA

```
//FIBIEN02 EXEC PGM=IBMMENSA,PARM='FI BE SUBJOB-2'  
//DD1 DD DSN=CAMPB.IEGW7990(+1),DISP=(OLD,CATLG),  
// UNIT=(,DEFER)  
//FINMAL02 EXEC PGM=IBMMENSA,PARM='FI MAL SUBJOB-2',COND=ONLY
```

IDCAMS

(Ver descripción y ejemplos en el apartado VSAM)

IEHLIST

IEBGENER

```
//P042 EXEC PGM=IEBGENER  
//OUT1 OUTPUT DEFAULT=YES,COPIES=1,FORMDEF=A10110  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=ADF.ADFM1P0.P040.DCF.LIST,DISP=(OLD,DELETE)  
//SYSUT2 DD SYSOUT=(D,,1549),OUTPUT=  
//SYSIN DD DUMMY,DSN=,DISP=  
//ABEND042 EXEC PGM=IBMABEND,COND=(0,EQ,P042)
```

IEBCOPY

Utilizado mayormente en la copia de ficheros particionados (PDS). Permite

- Copia de un PDS a secuencial
- Copia de uno o mas secuenciales a un PDS
- Copia o fusión de varios PDS
- Compresión de un PDS copiándolo sobre si mismo

Por cuestiones de integridad se aconseja el uso de SISCOPY en lugar del IEBCOPY

```
//P010 EXEC PGM=IEBCOPY  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=AJB.AJB00000.BACKUP.&DATAP,DISP=(SHR) ← Entrada  
//SYSUT2 DD DSN=&DSNIN,DISP=(SHR) ← Salida  
//SYSUT3 DD SPACE=(09076,(003750,003750),,,),AVGREC=U,  
// UNIT=(SYSALLDA,01)  
//SYSUT4 DD SPACE=(09076,(003750,003750),,,),AVGREC=U,  
// UNIT=(SYSALLDA,01)  
//SYSIN DD *  
COPY O=SYSUT2,I=((SYSUT1,R))  
/*
```

Ficheros de trabajo

IEBCOMPR

IEFBR14

El cometido de este programa es el de tan solo pasar código de retorno de control .

Ello es útil para efectuar tratamiento de ficheros como son el

CATALOGADO y BORRADO

```
//P005 EXEC PGM=IEFBR14
//DD1 DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,
// DISP=(MOD,DELETE,DELETE)
//P010 EXEC PGM=SORT
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=(SHR)
//SYSOUT DD SYSOUT=*
//SYSIN DD DSN=EXPLOT.@BHE.CONTROL(VCA04201),DISP=(SHR)
/*-----
/*- SORT FIELDS=(56,03,BI,A,59,05,BI,A)
/*-----
//SORTWK01 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK02 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK03 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK04 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK05 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK06 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
/*----- MOVIMIENTOS DE VENTAS Y ANULACIONES DEL DIA
//SORTIN DD DSN=@BHE.VCA.VCA00007.&DATAP,DISP=(SHR)
/*----- VENTAS/ANULACIONES DEL DIA ORDENADO POR OFICINA
//SORTOUT DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,DISP=(,PASS),
// SPACE=(01000,(015000,003000),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01),
// DATACLAS=,STORCLAS=,MGMTCLAS=,
// DCB=(RECFM=VB,LRECL=01000,BLKSIZE=0)
//ABEND010 EXEC PGM=IBMAEND,COND=(0,EQ,P010)
/*-----
/*- CATALOG DE LOS FICHEROS DEL PASO 10
/*-----
//P011 EXEC PGM=IEFBR14
//DD1 DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,
// DISP=(OLD,CATLG),UNIT=(,DEFER)
//ABEND011 EXEC PGM=IBMAEND,COND=(0,EQ,P011)
```

Espera la finalización correcta del paso anterior y si es así CATALOGA el fichero

Si el fichero existe lo borra/ Si no existe lo asume NEW y finaliza el paso de forma correcta

ISRSUPC

Utilidad dedicada a la búsqueda de uno o varios textos especificados en uno o mas miembros de un fichero particionado

```
//SEARCH EXEC PGM=ISRSUPC,
//      PARM=(SRCHCMP,
//      'ANYC')
//NEWDD DD DSN=EXPLOT.CONTROL,
//      DISP=SHR
//OUTDD DD DSN=T5152.U14.LLISTA,
//      DISP=OLD
//SYSIN DD *
SRCHFOR 'CLUSTER'
SRCHFOR 'REPRO'
SRCHFOR 'LISTCAT'
SRCHFOR 'IDCAMS'
SELECT
AAE01A01,AAE0200A,AAE0200B,AAE0200C,AAE0200D,AAE0200E
SELECT AAE0200F,AAE0200G,AAE0200H,AAE0200I,AAE0200J,AAE0200K
SELECT AAE02001,AAE02002,AAE02003,AAE02004,AAE02005,AAE02006
/*
```

} BÚSQUEDA

} MIEMBROS

IKJEFT01

```
/*-----**
/** P005 - DELETE TAULA ACIT02
/*-----**
/*@INI@ INCLUDE MEMBER=@INI@
//P005 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=P,HOLD=YES,FCB=6666
//SYSTSIN DD DSN=EXPLOT.CONTROL(ACI87102),DISP=(SHR)
//SYSIN DD *
DELETE FROM DBE2.ACIT02
COMMIT;
/*
DSN SYSTEM(DBE2)
RUN PROGRAM(DSNTEP2) -
LIB('DB2EXP2.RUNLIB.LOAD')
```

SISCOPY

```
//P030 EXEC PGM=SISCOPY
//SYSUT1 DD DSN=ACA.ACA07083,DISP=(SHR)
//SYSUT2 DD DSN=ACA.ACA42662(+1),DISP=(,PASS),
//      SPACE=(00080,(000100,000300,5),RLSE,,),AVGREC=U,
//      UNIT=(SYSALLDA,01),
//      DATACLAS=,STORCLAS=ECOPIA,MGMTCLAS=,
//      DCB=()
//SYSUT3 DD SPACE=(08906,(000178,000534),,,),AVGREC=U,UNIT=(SYSALLDA,01)
//SYSUT4 DD SPACE=(08906,(000178,000534),,,),AVGREC=U,UNIT=(SYSALLDA,01)
//SYSIN DD *
C O=SYSUT2,I=((SYSUT1,R)),LIST=NO
/*
//SYSPRINT DD SYSOUT=*
```

SISGENER

```
//P010 EXEC PGM=SISGENER
//SYSPRINT DD SYSOUT=*
//*_
//I001 DD DSN=EXPLOT.FONTS.ASM(TAUM7730),DISP=SHR
//O001 DD SYSOUT=P
//OUT1 OUTPUT DEFAULT=YES,DEST=R289,PAGEDEF=V88,COPIES=1
//*_
```

SISPOPCL

Programa de utilidad utilizado para la creación de ficheros vacíos. Tan solo realiza el OPEN y CLOSE del fichero especificado en la DD

```
//P010 EXEC PGM=SISPOPCL
//DD1 DD DSN=VCA.VCAD042.P010.&DATAP,DISP=(,PASS),
// SPACE=(01000,(015000,003000),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01),
// DATACLAS=,STORCLAS=,MGMTCLAS=,
// DCB=(RECFM=VB,LRECL=01000,BLKSIZE=0)
```

SISPTOOL

Permite efectuar copia de uno o varios ficheros en otros . Lo hace por llamada a la utilidad ICTOOL del DFSORT. Permite la codificación de varias DD s. I00n con O00n

```
//P030 EXEC PGM=SISPTOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//I001 DD DSN=@BHE.VCA.VCA00016(0),DISP=(SHR)
// DD DSN=@BHE.VCA.VCAD042.P020.&DATAP,DISP=(SHR)
//O001 DD DSN=@BHE.VCA.VCA00016(+1),DISP=(,PASS),
// SPACE=(00200,(000578,001734),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01),
// DATACLAS=,STORCLAS=,MGMTCLAS=,
// DCB=(MODEL,RECFM=VB,LRECL=00200,BLKSIZE=0)
```

PUTPARM

Programa utilizado para la incorporación de parámetros a un fichero temporal. Se recomienda su utilización para la incorporación de datos a programas del tipo BMP que trabajan en IMS.

```
//P010 EXEC PGM=PUTPARM,PARM='/&OPCIO&DATAD'
//SYSPRINT DD SYSOUT=*
//OUTPARM DD DSN=T5152.U14.LLISTA.&DATAP,DISP=(,PASS),
// SPACE=(00080,(000005,000015),,,),AVGREC=U,
// UNIT=(SYSALLDA,01),
// DATACLAS=,STORCLAS=,MGMTCLAS=,
// DCB=(LRECL=00080,RECFM=FB,BLKSIZE=0)
//ABEND010 EXEC PGM=IBMABEND,COND=(0,EQ,P010)
```

SORT

Este programa de utilidad esta pensado y optimizado para la clasificación de registros pero también realiza otras funciones tales como la selección de registros, modificación de los mismos, copia de ficheros y otras mas funciones que veremos.

Las fichas de control que utiliza son:

SORTLIB

Librería de la instalación donde reside el ejecutable del programa SORT

SORTIN

Fichero con los registros de entrada para su posterior clasificación. En el caso de MERGE deberán separarse cada uno de ellos con un numero de SORTIN (SORTIN01,SORTIN02.....)

SORTOUT

Fichero donde ubicaremos el resultado de la clasificación de salida. No será preciso especificar la DCB del fichero de salida puesto que tomará las mismas características que el de entrada.

SORTWKnn

Son ficheros de trabajo que utiliza el sistema para guardar y clasificar los registros de entrada lo hace en disco por lo que será siempre UNIT=SYSALLDA

SYSIN

Esta ficha de control no es especifica del SORT pero su aplicación en esta utilidad sirve para especificar las sentencias de control del SORT. Por lo general los diferentes ficheros que utiliza son guardado en una librería particionada con los datos de cada SORT

```
//SYSIN DD DSN=EXPLOT.@BHE.CONTROL(VCA04201),DISP=(SHR)
```

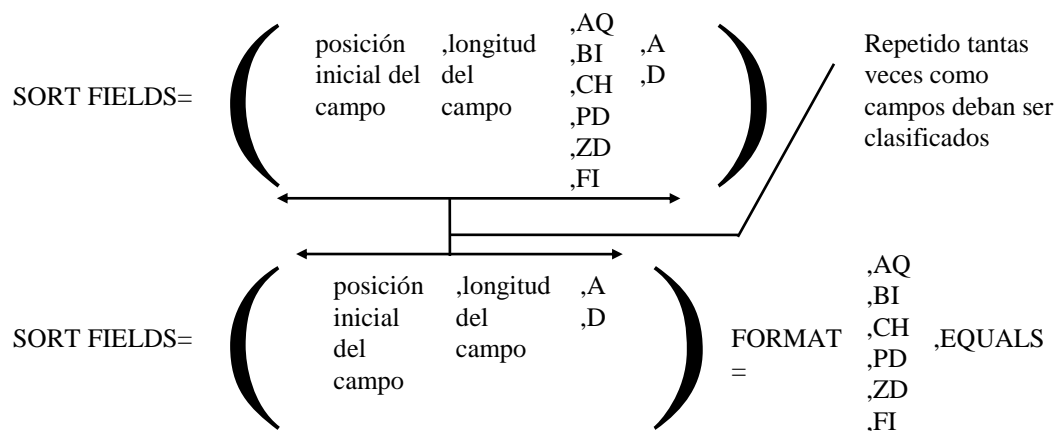
```

/*****
/**** SORT MOVIMIENTOS DE VENTAS/ANULACIONES POR OFICINA ****
/*****
/*_
//P010 EXEC PGM=SORT
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=(SHR)
//SYSOUT DD SYSOUT=*
//SYSIN DD DSN=EXPLOT.@BHE.CONTROL(VCA04201),DISP=(SHR)
/*-----
/*- SORT FIELDS=(56,03,BI,A,59,05,BI,A)
/*-----
//SORTWK01 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK02 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK03 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK04 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK05 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
//SORTWK06 DD SPACE=(08906,(001098,000366),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01)
/*----- MOVIMIENTOS DE VENTAS Y ANULACIONES DEL DIA
//SORTIN DD DSN=@BHE.VCA.VCA00007.&DATAP,DISP=(SHR)
/*----- VENTAS/ANULACIONES DEL DIA ORDENADO POR OFICINA
//SORTOUT DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,DISP=(,PASS),
// SPACE=(01000,(015000,003000),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01),
// DATACLAS=,STORCLAS=,MGMTCLAS=,
// DCB=(RECFM=VB,LRECL=01000,BLKSIZE=0)
//ABEND010 EXEC PGM=IBMAEND,COND=(0,EQ,P010)
/*-----
/*- CATALOG DE LOS FICHEROS DEL PASO 10
/*-----
//P011 EXEC PGM=IEFBR14
//DD1 DD DSN=@BHE.VCA.VCAD042.P010.&DATAP,
// DISP=(OLD,CATLG),UNIT=(,DEFER)
//ABEND011 EXEC PGM=IBMAEND,COND=(0,EQ,P011)

```


SORT

La sentencia de control permite clasificar los registros según los parámetros que se le especifican.



SORT FIELDS= COPY

AQ Siempre que haya una sentencia ALTSEQ que modifique el peso binario de clasificación

BI Binario (acepta la clasificación de todos los campos salvo los empaquetados(PD-ZD) con signo cuando ellos deben ser clasificados colocando primero los negativos y luego los positivos o en el sentido contrario (-1,-5,+1,+2) ya que en binario asociaría la clasificación por el numero y en números iguales el signo. (+1,-1,+2,-5)

CH Carácter (alfanumérico) _

PD Empaquetado

ZD numérico con signo

FI coma flotante

A Ascendente (blanco,a.....z,A.....Z,0....9) Hexadecimal -> (40,81....A9,C1...E9,F0....F9) _

D Descendente (9....0,Z....A,z....a,blanco) Hexadecimal -> (F9.....F0,E9....C1,A9.....81,40)

FORMAT

Este parámetro puede suplir el posicional de formato cuando solo es un campo a clasificar o son varios de igual formato

EQUALS

En claves duplicadas se mantendrá el mismo orden de recepción de Input a Output. Por lo general esta opción es la que se mantiene en las instalaciones por defecto.

COPY

Con SORT FIELDS= COPY evitamos el clasificar los registros y efectuamos copia de los mismos tal y como aparecen de entrada.

Produce iguales resultados que MERGE FIELDS = COPY o que OPTION = COPY

```
//SYSIN DD *
SORT FIELDS=(1,75,BI,A)
/*
```

Nota: En fichero de longitud variable habremos de sumar las 4 posiciones que encabezan el registro a la longitud inicial de cada campo

SENTENCIAS DE CONTROL(SORT)

Son parámetros que precisa el programa SORT y que se especifican en la ficha SYSIN para determinar cuales son los criterios de clasificación, la incorporación de datos, la exclusión de datos o registros, orden de clasificación y otros datos precisos para realizar el trabajo como realmente queremos.

La forma en rellenar la sentencia de control parte de la segunda columna del miembro particionado hasta la columna 71. Tan solo podrá codificarse la primera columna cuando se trate de un comentario y en ella deberá situarse un asterisco (*) (El resto de la fila se toma como comentario). Si la especificación de la ficha de control supera el espacio de una fila deberemos ocupar la columna 72 con cualquier carácter no blanco y proseguir en la línea siguiente comenzando en la columna 16. Son sentencias de control del Sort:

ALTSEQ

Sirve para alterar el orden de clasificación del SORT. Basándose en el valor hexadecimal y sin modificarlo, tan solo rectificando su peso binario

Clasificar la Ñ entre N y O

```
//SYSIN DD *
SORT FIELDS=(1,75,AQ,A)
ALTSEQ CODE=(7BD6,D6D7,D7D8,D8D9,D9DA)
/*
```

valor real Ñ
peso binario =O
valor real O
peso binario =P
valor real P
peso binario =Q
valor real Q
peso binario =R
valor real R
peso binario =(código sin valor EBCDIC)
El formato ha de ser obligatoriamente AQ

INCLUDE

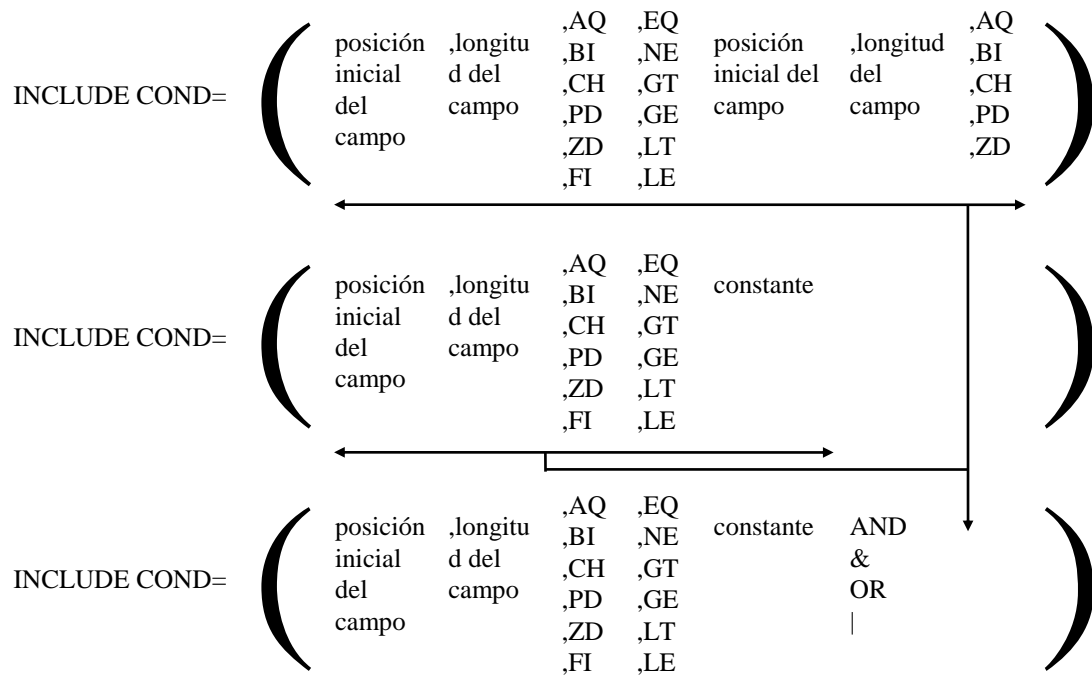
Permite efectuar una previa selección de los registros a clasificar cuando cumplen un determinado valor para uno o varios campos, ignorando la salida de los que no la cumplen.

Pueden compararse:

Campos con otros campos del mismo registro

Campos con valores constantes

Una o varias comparaciones con operadores lógicos



EQ Igual

NE No igual

GT Mayor que

GE Mayor o igual que

LT Menor que

LE Menor o igual que

constante

C'.....' Constante en formato carácter

X'.....' Constante en formato hexadecimal

B'.....' Constante binaria

0.....9 Constante numérica

Operador lógico

AND Y

& Y

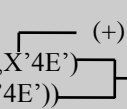
OR O

| O

En los operadores lógicos prevalece el AND antes que el OR pero en cualquier caso es buena costumbre el cerrar las operaciones lógicas con paréntesis cuando sean varias las operaciones a realizar

```
//SYSIN DD *
SORT FIELDS=(1,75,CH,A)
INCLUDE COND=(1,75,CH,EQ,76,75,CH)
/*
```

```
INCLUDE COND=(1,75,CH,EQ,76,75,CH)
INCLUDE COND=(1,75,,EQ,76,75) ,FORMAT=CH
INCLUDE COND=(166,4,BI,GT,162,4,BI)
INCLUDE COND=(5,8,GT,13,8,OR,105,4,LE,1000),FORMAT=CSF
INCLUDE COND=(5,4,CH,EQ,C'DEBE',AND,21,8,ZD,GT,+1000,OR,31,1,CH,NE,X'4E')
INCLUDE COND=(5,4,CH,EQ,C'DEBE',&(21,8,ZD,GT,+1000,OR,31,1,CH,NE,X'4E'))
```



Los resultados serán diferentes por el orden de los operadores lógicos

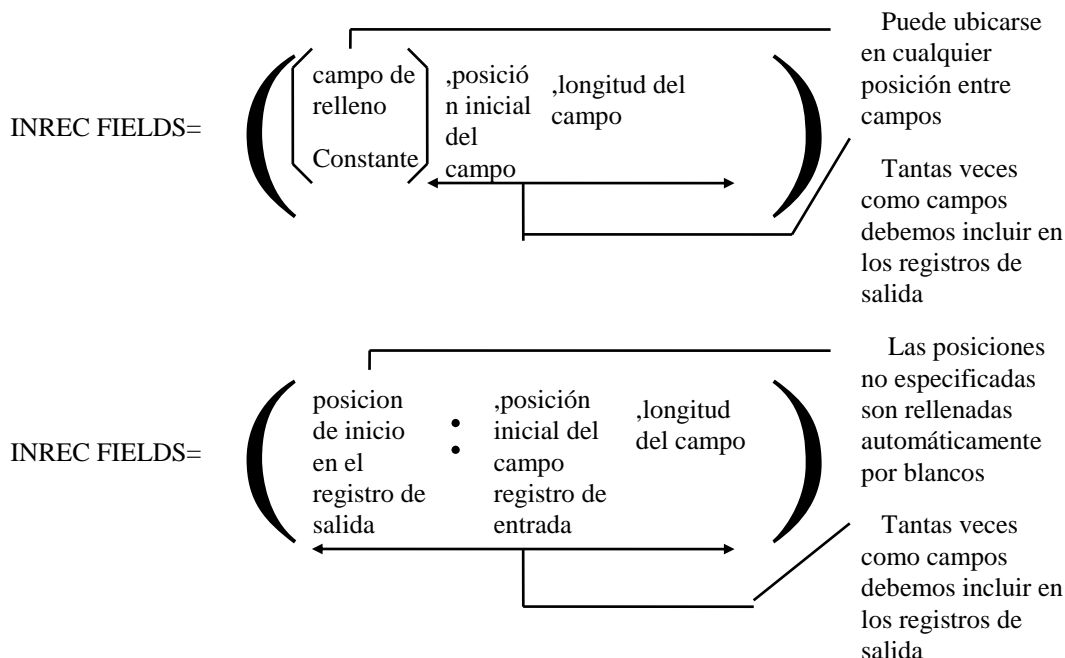
OMIT

La forma de trabajar de esta sentencia de control es parecida a la anterior INCLUDE salvo que mientras en ella cuando se cumplían los resultados era un registro a incorporar a la salida en el caso de OMIT el cumplirse la condición presupone el despreciar el registro y prescindir de el en salida

```
//SYSIN DD *
SORT FIELDS=(1,75,CH,A)
OMIT COND=(1,75,CH,EQ,76,75,CH)
/*
```

INREC

Modifica los registros de entrada antes de clasificarlos con SORT. Por medio de INREC pueden seleccionarse cuales serán los campos que traspasaremos a los registros de salida, ignorando los restantes e incluso podemos añadir nuevos campos con un valor constante



Campo de relleno	Puede estar situado en cualquier posición (inicio,final,n seguidos,entre campos,.....). Especifican el tipo de relleno:
X,nX	Relleno de 1 o n blancos
Z,nZ	Relleno de 1 o n ceros binarios
Constante	Puede estar situado en cualquier posición (inicio,final,n seguidos,entre campos,.....). Determinan el tipo de constante:
X'....',nX'....'	Formato hexadecimal (multiplo de 2).Repetido n veces
C'....',nC'....'	Caracteres. Repetido n veces
	Si tenemos que aplicar una (') entre constantes deberemos aplicar doble ('') y solo contara como una

FORMATO	EN REGISTRO DE SALIDA
INREC FIELDSD=(10,3,20,8)	De 1 a 3 contenido entrada 10 a 12 De 4 a 11 contenido entrada 20 a 27
INREC FIELDSD=(10,3,3X,20,8)	De 1 a 3 contenido entrada 10 a 12 De 4 a 6 blancos De 7 a 14 contenido entrada 20 a 27
INREC FIELDSD=(20:C'TOTAL ')	De 1 a 19 blancos De 20 a 25 TOTALb
INREC FIELDSD=(1,5,10:C'TOTAL ',16:5,10,2X'FF')	De 1 a 5 contenido entrada de 1 a 5 De 6 a 9 blancos De 10 a 14 TOTAL 15 a blanco De 16 a 25 contenido entrada de 5 a 14 26 el formato hexadecimal 'FFFF'

```
//SYSIN DD *
INREC FIELDS=(10,3,20,8,33,11,5,1)
SORT FIELDS=(4,8,CH,A,1,3,BI,A)
/*
```

Los campos a clasificar serán los del registro de salida que se corresponden al campo que se inicia en la posición 20 de 8 posiciones y el que se inicia en la posición 10 de 3 posiciones del registro de entrada. EN INREC LA CLASIFICACION ES POSTERIOR A LA SELECCIÓN DE CAMPOS

OUTREC

Esta sentencia de control se rige por los mismos parámetros que INREC. La diferencia entre una y otra se basa en que (INREC) modifica los registros antes de la clasificación de salida mientras que (OUTREC) Lo hace una vez estos han estado clasificados

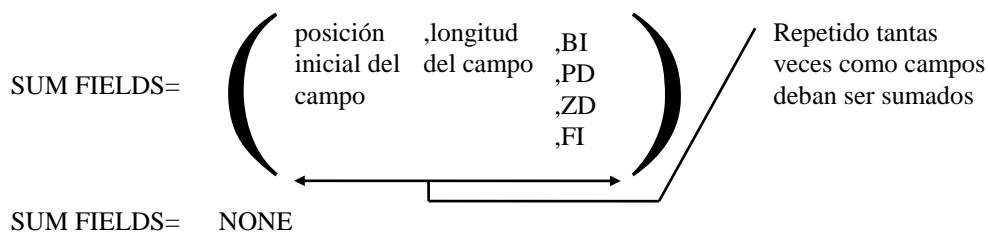
```
//SYSIN DD *
SORT FIELDS=(4,8,CH,A,1,3,BI,A)
OUTREC FIELDS=(4,8,1,3,12,5)
/*
```

MERGE

Esta sentencia de control permite juntar registros de dos o mas ficheros en un único fichero de salida (Matching) ordenados por unos determinados campos. Se basa en los mismos parámetros que el SORT para ordenar los registros en el fichero de salida.

SUM

Con la sentencia de control SUM podemos efectuar los totales de campos numéricos de los registros que son coincidentes con los criterios de clasificación de la sentencia SORT o MERGE. De tantos registros duplicados por clave de clasificación como hayan tan solo en salida se gravará uno y el campo especificado en SUM contendrá la suma de ese campo numérico de todos los registros duplicados. Así pues una de las primeras finalidades de la sentencia sum se basa en evitar claves duplicadas y adicionalmente el sumar campos numéricos de igual clave



NONE

Evita las claves duplicadas sin necesidad de sumar ningún campo numérico

OPTION

La sentencia de control OPTION permite alterar las opciones que tiene por defecto el SORT o MERGE. Son tantas las posibilidades como posibles opciones tienen ambas pero de todas ellas vamos a citar las de mayor utilidad como son:

COPY

La opción COPY produce iguales resultados que SORT=COPY o MERGE=COPY permite sin necesidad de clasificar el registro de entrada y guardando el mismo orden de recepción de los datos utilizar otras sentencias de control como son INCLUYE , OMITA, INREC, OUTREC, SKIPREC, STOPAFT y otras . Se excluye la sentencia SUM que precisa los criterios de clasificación para realizar las agrupaciones de claves iguales y totales.

NOEQUALS

EQUALS

En aquellos casos en que coincide el criterio de clasificación del SORT o MERGE se mantiene el mismo orden de recepción de estos registros. Lo contrario será el NOEQUALS que puede alterar ese orden que proviene del fichero de entrada.

NOLIST

LIST

Lista las sentencias de control , o en el caso de NOLIST prescinde el hacerlo.

SKIPREC

La opción SKIPREC=n permite saltar un numero determinado de registros del principio. Serán pues registros que no formaran parte de los registros del fichero de salida.

STOPAFT

Esta opción permite parar la selección de registros de salida una vez el numero de ellos sea coincidente con el valor expresado en STOPAFT=n. Si la sentencia se asocia con otras que limitan o excluyen registros de salida como son SKIPREC,INCLUDE,OMIT, la opción STOPAFT dará salida como máximo a tantos como cumplan esas condiciones y ello puede suponer un numero mayor de registros de entrada para que se cumpla ese limite de registros en salida.

La mayoría de las sentencias de control pueden ser mezcladas para la obtención correcta del resultado final vemos aquí algunos ejemplos de diferentes combinaciones.

```
//SYSIN DD *  
OMIT COND=(19,4,CH,EQ,23,4,CH)  
INCLUDE COND=(19,4,CH,EQ,C'9411')  
SORT FIELDS=(19,8,BI,A),EQUALS  
OPTION SKIPREC=5.STOPAFT=1000  
/*
```

Se incluyen los registros que los cuatro caracteres que ocupan la posición 19 sean iguales a 9411 pero de estos prescindiremos de los que este valor coincida con los 4 caracteres de la posición 23
Saltará 5 registros de entrada y dará salida como máximo a un total de 1000 registros que cumplan las condiciones citadas
Si existe duplicidad en las 8 posiciones que parten de la posición 19 se mantendrá el mismo orden que tenían estos registros en el fichero de entrada.

```
//SYSIN DD *  
INCLUDE COND=(20,8,CH,EQ,C'ALMACEN ')  
OMIT COND=(33,11,CH,EQ,C'GRANADA')  
INREC FIELDS=(10,3,20,8,33,11,5,1)  
SORT FIELDS=(12,11,CH,A,1,3,BI,A)  
SUM FIELDS=(1,3,BI)  
/*
```

Escogeremos aquellos que sus 8 caracteres que parten de la posición 20 sean iguales a ALMACEN, pero prescindiremos dentro de estos de los que sus 11 caracteres que se encuentran en la posición 33 sean iguales a GRANADA.
Tan solo precisamos un registro por almacén en las 11 posiciones que parten de la 12 posición del registro de salida, con el total de existencias que tenemos en las tres primeras posiciones.

VSAM

Vsam se corresponde al acronimo de *Virtual Storage Access Method*. (Método de acceso a la memoria virtual). Se basa en la organización de ficheros y método de acceso de alto rendimiento y ello lo realiza mediante una estructura de catálogos. Utiliza el concepto de memoria virtual. Son ficheros de alta fiabilidad que garantizan la seguridad e integridad de los datos y a los que se puede acceder de forma rápida.

VSAM gestiona dos partes de los ficheros de este tipo como son el CATALOGO y los DATOS. En la parte correspondiente al CATALOGO, obtenemos la información precisa para el definir y acceder al fichero como son (NOMBRE DEL FICHERO, TAMAÑO DEL REGISTRO, ESPACIO LIBRE.....)

La asignación de espacio de los ficheros VSAM se basa en las AREAS DE CONTROL que son la agrupación de un grupo de INTERVALOS DE CONTROL y cada uno de estos INTERVALOS DE CONTROL lo integran unos campos que aportan información del intervalo y datos como son ubicación y espacio libre (El campo se denomina CDF y tan solo puede aparecer uno en cada Intervalo de control). Existe a su vez otro campo que describe la longitud de los registros y determina el numero de registros adyacentes de igual longitud (el campo se denomina RDF y pueden aparecer tantos como espacio contenga el intervalo de control) Aparecen después los registros lógicos con la información guardada y por lo general y con mayor o menor amplitud aparece el espacio libre que nos permitirá insertar nueva información al fichero

Tipos de ficheros VSAM:

KSIDS

Son ficheros ordenados en secuencia de claves. Los registros son recuperados de forma secuencial o directamente por su clave que será exclusiva, contigua y en la misma posición en cada uno de los registros. Pueden ser de longitud fija o variable y la carga de los datos debe realizarse en orden secuencial ascendente por el numero de clave.

El acceso a los datos se realiza por medio de los índices. La estructura de los índices se basa igualmente en intervalos de control y áreas de control. Un área de control de datos es apuntada por un único intervalo de control de índices.

ESDS

Son ficheros ordenados en secuencia de entrada. La carga se efectúa de forma secuencial y se añaden registros al final del fichero pero en tanto la carga como en la adición de nuevos ficheros se guarda la posición de inicio de cada registro por lo que el acceso puede hacerse de forma secuencial o bien por el RBA *Relative Byte Address* o Dirección Relativa de Octeto. Los registros pueden ser de longitud variable y se permiten modificaciones en los registros de un ESDS siempre que ellas no modifiquen la longitud del registro.

RRDS

Son ficheros de registros relativos. La división de registros en un fichero del tipo RRDS se realiza por medio de espacios idénticos lo que supone en el caso de la supresión de un registro que se creen huecos que podrán rellenarse mas tarde con la nueva incorporación de otros registros. La recuperación de un registro del tipo RRDS se efectúa por medio del numero de registro relativo que se corresponde al numero de cada hueco de registro

LDS

Fichero de tipo lineal. Los ficheros del tipo LDS no contienen información de control y el acceso es únicamente por medio del intervalo de control.

INDICE CLUSTER

Es la clave única de acceso a un fichero del tipo KSDS o ESDS ordenada de forma ascendente. Solo puede existir un índice cluster por tipo de fichero VSAM.

INDICE ALTERNATIVO (AIX)

Es aquel que permite el acceso directo a los datos por un orden diferente al del índice cluster. Tiene igual estructura que la que se corresponde a un fichero KSDS con datos e índices y se ordenan en forma ascendente por numero de clave alternativa. Un índice alternativo puede crearse para ficheros del tipo KSDS y ESDS. No existe limite por lo que a la creación de índices alternativos se refiere.

PATH

Es la relación lógica que existe entre un INDICE CLUSTER y un INDICE ALTERNATIVO

IDCAMS

Es el nombre del modulo que invoca a un conjunto de funciones útiles para establecer y mantener catálogos y ficheros VSAM. Las funciones que le son permitidas al IDCAMS son:

- Definir
- Copiar
- Listar
- Imprimir
- Comprobar
- Borrar

Cada una de estas funciones son solicitadas por medio de sentencias de control que son expresadas después de la columna 1 que precisan de un (-) si quieren continuarse en una nueva línea. Se permiten múltiples sentencias dentro de una misma ejecución de un IDCAMS

```
//P010 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  REPRO INFILE(IN) OUTFILE(OUT) _____
/*
//IN DD DSN=T9745.VCAORIG.D6801.D6919,DISP=(OLD)
//OUT DD DSN=@BHE.VCA.VCA00005.&DATAP,DISP=(,PASS),
// SPACE=(00500,(000200,000600),RLSE,,),AVGREC=U,
// UNIT=(SYSALLDA,01),
// DATACLAS=,STORCLAS=BROWSET,MGMTCLAS=,
// DCB=(RECFM=VB,LRECL=0500,BLKSIZE=0)
//TOOLMSG DD SYSOUT=*
```

Sentencia de Control

DEFINICIÓN DE VSAM

Para crear un fichero VSAM es preciso crear las tres partes que lo componen:

CLUSTER

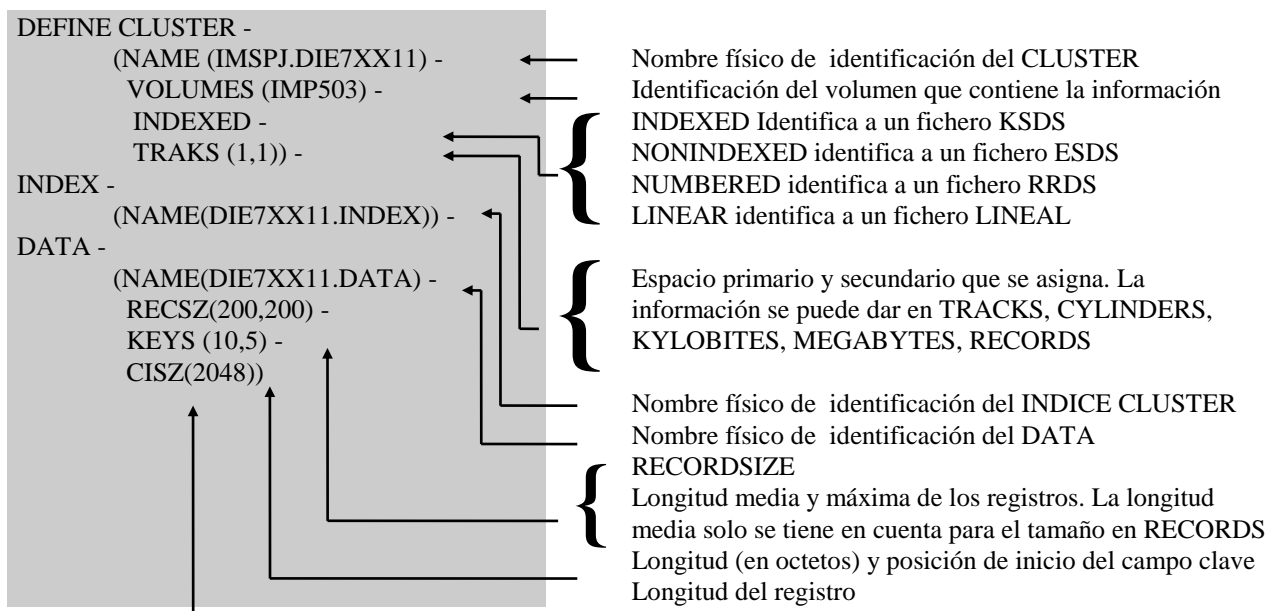
Esta parte aporta información del fichero referente a nombre, ubicación, espacio y demás características físicas..

DATA

Destinada a contener los datos del fichero.

INDEX

Esta división hace referencia a la información de índice para el acceso a los datos. Hay ficheros de tipo VSAM que no requieren acceso directo y prescinde de este apartado.



BORRADO DE VSAM

El borrado de un fichero VSAM presupone el borrado de todo lo relacionado con el CLUSTER, DATA, INDICE CLUSTER, INDICES ALTERNATIVOS (AIX), PATH,

DELETE

Es la sentencia que determina el borrado del fichero

CLUSTER

Indica que el borrado se relaciona con un VSAM

PURGE

Da autorización al borrado del fichero todo y sin haberse cumplido el plazo de su vencimiento

ERASE

Indica que la parte de datos relacionada se sobrescriba con 0 binarios

```
DELETE -  
(IMSPJ.DIE7XX11) PURGE CLUSTER ERASE  
  ↑ Nombre físico de identificación CLUSTER del fichero a borrar
```

COPIADO DE VSAM

La utilidad que permite efectuar copias de un fichero VSAM permite a su vez realizarlo con otro tipo de ficheros de diferente tipo de organización. Son parámetros asociados REPRO, INFILE, OUTFILE, INDATASET, OUTDATASET, COUNT, SKIP

REPRO

La sentencia determina el copiado del fichero

INFILE

INDATASET

Da a conocer el nombre del fichero de entrada

OUTFILE

OUTDATASET

Da a conocer el nombre del fichero de salida

FROMKEY

TOKEY

En fichero KSDS permite especificar la clave de inicio y la de final

FROMADDRESS

TOADDRESS

En fichero ESDS permite especificar la Dirección Relativa de Octeto (RBA) de inicio y la de final

FROMNUMBER

TONUMBER

En fichero RRDS permite especificar el numero de Registro Relativo de inicio y la de final

COUNT

Numero de registros a copiar

SKIP

Numero de registros a saltar

REPRO -

INFILE (IMSPJ.DIE7XX11) OUTFILE (IMSPJ.DIE7XX11.V2) SKIP(100) COUNT(50)

Numero de ficheros que saltara de inicio sin copiar

Numero de registros que copiará

IMPRESIÓN DE VSAM

Podremos imprimir ficheros del tipo VSAM e incluso de otros tipos, total o parcialmente por medio de los parámetros PRINT, CHAR, INFILE, INDATASET, COUNT, SKIP

PRINT

La sentencia determina la impresión del fichero

INFILE

INDATASET

Da a conocer el nombre del fichero de entrada

CHAR

Permite que el formato de salida sea en un formato legible . Si no se especifica este parámetro la impresión es en formato DUMP

FROMKEY

TOKEY

En fichero KSDS permite especificar la clave de inicio y la de final

FROMADDRESS

TOADDRESS

En fichero ESDS permite especificar la Dirección Relativa de Octeto (RBA) de inicio y la de final

FROMNUMBER

TONUMBER

En fichero RRDS permite especificar el numero de Registro Relativo de inicio y la de final

COUNT

Numero de registros a copiar

SKIP

Numero de registros a saltar

PRINT -

INFILE (IMSPJ.DIE7XX11) CHAR -
SKIP(100) COUNT(50)

Numero de registros que copiará

Numero de ficheros que saltara de inicio sin copiar

LISTADO DE CATALOGO DE VSAM

Podremos imprimir el Listado del catalogo de un fichero VSAM, total o parcialmente por medio de los parámetros LEVEL, ENTRIES, ALL, VOLUME

LISTCAT

La sentencia determina el listado del catalogo

LEVEL

Imprime todos los que parten de un determinado nivel de nombre especificado. Si por ejemplo especificamos “ENTRADA”, nos dará lista de “ENTRADA.F120197” y “ENTRADA.F130197”

ENTRIES

Lista el contenido de un fichero con el nombre especificado.

ALL

Determina que hay que imprimir toda la información del catalogo.

VOLUME

En un fichero multivolumen tan solo aporta la información que se corresponde al volumen especificado.

NAME

Es un parámetro asumido por defecto asociado a ENTRIES o LEVEL

```
LISTCAT -  
ENTRIES(IMSPJ.DIE7XX11) ALL
```

DEFINICIÓN DE UN AIX (ÍNDICE ALTERNATIVO)

Para crear un Índice Alternativo de un fichero VSAM es preciso crear las tres partes que lo componen:

DEFINE AIX

Esta parte aporta información del INDICE referente a nombre, ubicación, espacio y demás características físicas..

DATA

Destinada a contener los datos del índice del fichero.

INDEX

Esta división hace referencia a la información de índice para el acceso a los datos.

DEFINE AIX -		
(NAME (IMSPJ.DIE7XX11.AIX) -	←	Nombre físico de identificación del AIX
RELATE (IMSPJ.DIE7XX11) -	←	Nombre del CLUSTER base con el que esta relacionado
VOLUMES (IMP503) -	←	Identificación del volumen que contiene la información
NONUNIQUEKEY -	←	NONUNIQUEKEY Clave(KSDS-ESDS) puede ser repetida
TRAKS (1,1) -	←	
RECSZ (1024,2572) -	←	UNIQUEKEY Clave(KSDS-ESDS) no puede ser repetida
KEYS (7,32) -	←	Longitud media y máxima cuando se mide en RECORDS
INDEX -		Longitud y posición relativa de inicio (1ª posición = 0)
(NAME(DIE7XX11.AIX.INDEX)) -		
DATA -		
(NAME(DIE7XX11.AIX.DATA)		

CARGA DE UN AIX (INDICE ALTERNATIVO)

Para cargar un Índice Alternativo de un fichero VSAM es necesario que previamente se haya hecho la carga del índice CLUSTER dado que el AIX se carga partiendo de los datos de ese índice CLUSTER.

BLDINDEX

La sentencia determina la carga del AIX

INDATASET

Nombre del CLUSTER base (KSDS o ESDS)

OUTDATASET

Nombre del AIX

```
BLDINDEX -  
  INDATASET (IMSPJ.DIE7XX11) -  
  OUTDATASET (IMSPJ.DIE7XX11.AIX)
```

—&—
&&, 33

—*—
*, 28

—/—
/*, 9; 10
//, 9; 10
//*, 9; 10

—A—
ABENDED, 45
ABSTR, 37
ACCT, 21
ADDRSPC, 15; 21
ADUEMAIN, 53
AFF, 36
AFINIDAD, 36
AIX, 72; 78
AL, 38
ALTSEQ, 65
ALX, 37
ALL, 77
AUL, 38

—B—
BLDINDEX, 78
BLKSIZE, 39
BLP, 38

—C—
CATLG, 34
CLASS, 14
CLUSTER, 72; 73; 74
COND, 15; 21
CONTIG, 37
COPIES, 31
COPY, 17; 64; 70
COUNT, 75; 76
CYL, 37
CHAR, 76

—D—
DA, 39
DATA, 28; 73; 78
DBA00111, 54
DBA00121, 54
DBA00211, 54
DBA00311, 54
DBA00321, 54
DCB, 39
DD, 25
DDNAME, 30
DEFER, 36
DEFINE AIX, 78

DELETE, 34; 74
DEN, 40
DEST, 31
DFSRR00, 55
DISP, 34
DITTO, 56
DPRTY, 22
DSN, 32
DSNAME, 32
DSNUTILB, 57
DSORG, 39
DUMMY, 29
DUPLICATE NAME, 44
DYNAM, 29
DYNAMNBR, 16; 22

—E—
ENDED, 45
ENTRIES, 77
EQUALS, 64; 70
ERASE, 74
ESDS, 71
EVEN, 21
EXEC, 20
EXPDT, 38

—F—
F, 39
FB, 39
FBA, 39
FBM, 39
FORMAT, 64
FREE, 31
FROMADDRESS, 75; 76
FROMKEY, 75; 76
FROMNUMBER, 75; 76

—G—
GDG, 33

—H—
HOLD, 17; 31

—I—
IBMABEND, 58
IBMMENSA, 58
IDCAMS, 72
IEBCOMPR, 58
IEBCOPY, 58
IEBGENER, 58
IEFBR14, 59
IEHLIST, 58
IKJEFT01, 60
IN, 38
INCLUDE, 66
INDATASET, 75; 76; 78
INDEX, 73; 78
INFILE, 75; 76
INREC, 68

IS, 39
ISRSUPC, 60

—J—

JCL ERROR, 44
JCLHOLD, 17
JES2, 8
JOB, 13
JOBCAT, 26
JOBLIB, 26
JOBPARM, 19

—K—

KEEP, 34
KSDS, 71

—L—

LABEL, 38
LDS, 71
LEVEL, 77
LINECT, 19
list, 70
LISTCAT, 77
LRECL, 39
LTM, 38

—M—

MERGE, 69
MOD, 34
MSGCLASS, 14
MSGLEVEL, 14
MXIG, 37

—N—

NAME, 77
NEW, 34
NL, 38
noequals, 70
nolist, 70
NONE, 69
NOPWREAD, 38
NOT FOUND, 44
NOTIFY, 14
NSL, 38

—O—

OLD, 34
OMIT, 67
ONLY, 21
OPTION, 70
OUT, 38
OUTDATASET, 75; 78
OUTFILE, 75
OUTLIM, 32
OUTREC, 69

—P—

PARM, 23

PASS, 34
PASSWORD, 38
PATH, 72
PDS, 33
PEND, 41
PERFORM, 18; 23
PGM, 20
PO, 39
PRINT, 76
PROC, 20; 41
PRTY, 16
PS, 39
PURGE, 74
PUTPARM, 61

—R—

RD, 17; 23
RECFM, 39
REGION, 24
REGIÓN, 18
REPRO, 75
RESTART, 17
RETPD, 38
RLSE, 37
ROUND, 37
ROUTE, 19
RRDS, 71

—S—

SCAN, 17
SHR, 34
SISCOPY, 60
SISGENER, 61
SISPOPCL, 61
SISPTOOL, 61
SKIP, 75; 76
skiprec, 70
SL, 38
SORT, 62; 64
SORTIN, 62
SORTLIB, 62
SORTOUT, 62
SORTWK, 62
SPACE, 37
STEPCAT, 27
STEPLIB, 27
stopaft, 70
SUL, 38
SUM, 69
SYSABEND, 27
SYSALLDA, 36
SYSCHK, 26
SYSIN, 62
SYSOUT, 30
SYSTEM CODES, 49
SYSUDUMP, 27
SYSWK, 36

—T—

TA80, 36
TA90, 36
TAPE, 36

TIME, 18; 24
TOADDRESS, 75; 76
TOKEY, 75; 76
TONUMBER, 75; 76
TRK, 37
TYPRUN, 17

—U—

U, 39
UCS, 32
UNCATLG, 34
UNIT, 36

USER CODES, 51

—V—

V, 39
VB, 39
VOLUME, 77
VSAM, 71

—W—

WRK80, 36