

## Utilidades MVS (III)

### Entradilla:

El presente artículo cierra esta miniserie dedicada a las utilidades del MVS. Son muchas e importantes las utilidades que se quedan en el tintero, pero el objetivo que se había marcado con este tema solo era comentar las 'utilities' de uso mas frecuente, asi como resaltar la idea de que un desarrollador debe disponer de forma fácil y accesible de todos los manuales de los productos instalados, sin necesidad de recurrir a la fotocopia de apuntes manuscritos de otro compañero.

### 1.- Introducción

Como todo informático conoce, el MVS es el sistema operativo para grandes procesadores de IBM, estando justificada la existencia de estos grandes procesadores por la necesidad de gestionar grandes volúmenes de datos.

En los dos artículos anteriores se han comentado algunas de las muchas utilidades estándar con que cuenta una instalación de este tipo, pues este S.O. se encuentra acompañado por un conjunto de programas que sin formar parte del mismo, facilitan las labores básicas de gestión de ficheros, tales como borrar, imprimir, copiar, etc., lo cual, por otra parte cabe comentar que es algo normal, y que ocurre con cualquier sistema operativo.

Las utilidades vistas hasta ahora, provienen incluso de la versión anterior del MVS, concretamente del OS/VS, pero que con objeto de mantener la migración de aplicaciones a las versiones nuevas, han evolucionado junto con él.

Los programas que se van a tratar en este artículo no son simples programas de utilidad como los vistos anteriormente, sino que son unas 'facilities' o utilidades integradas y más completas (aunque no mas complejas) pero que también son de gran utilidad para el desarrollador:

- **IDCAMS** : Es una utilidad definida inicialmente para manejar objetos VSAM, y que formando parte del DFP (Data Facility Product), permite utilizar todos los métodos de acceso, excepto los ficheros particionados.
- **IKJEFT01**: Es el propio **TSO** (Time Sharing Operation).
- **PRINTDS**: Comando de TSO para imprimir ficheros.
- **ISRSUPC**: Programa que permite buscar cadenas de caracteres entre los distintos miembros de una librería o fichero PDS.

### 2.- Versiones del MVS

Con el fin de enmarcar en el tiempo la evolución del **MVS** (Multiple Virtual Storage), se incluye este apartado que enumera las distintas versiones del mismo, resaltando el hecho de que son las diferentes versiones de los productos que acompañan al MVS básico las que establecen las diferentes versiones de este S.O.

Así, cuando se habla de:

-La **versión 1 de MVS (1972)**, con sus diferentes revisiones (releases), se esta hablando de lo que comúnmente se conoce como **MVS/370**, pero se corresponde con el **OS/VS1 versión 2**, e introduce el concepto de Espacio de direcciones.

-La **versión 2 de MVS (1980)**, más sus diferentes releases componen el **MVS/XA**. (Extended Architecture).

Esta realmente fue una revisión de la versión anterior para solucionar la gran cantidad de problemas presentados en aquella.

El MVS/XA es una versión que afecta tanto al hardware como al software ya que la memoria virtual pasa de 16 Mb a 2 Gb, necesitando 4 bytes para direccionar cada posición de memoria. (ver en num.18 de Solo programadores - Campos de la PSW : pag. 43)

-La **versión 3 de MVS (1988)**, se corresponde con lo que se conoce como **MVS/ESA** (Enterprise System Architecture).

Esta versión se diferencia de la anterior, entre otras cosas, en que puede direccionar hasta un máximo de 15 espacios de direcciones de 2 Gb. Así mismo esta versión introduce el uso de microcódigo.

En futuros artículos se comentarán los distintos conceptos que se mencionan en esta breve reseña histórica del MVS.

## 2.- IDCAMS

La 'Facility' IDCAMS (Access Method Services) es una utilidad proporcionada por IBM para controlar objetos VSAM, tales como catálogos, paths, clusters, índices alternativos, etc. (ver *Solo Programadores: Ficheros VSAM*), aunque también puede ser usado para manejar todos los tipos de ficheros a excepción de los PDS o ficheros particionados.

Desde el punto de vista del JCL no se diferencia en nada de las utilidades vistas, pues utiliza las mismas Ddnames SYSPRINT y SYSIN.

Los comandos admitidos por esta utilidad a través de la **SYSIN** se podrían clasificar en:

### 2.1.- Creación de catálogos:

Los ficheros VSAM no pueden ser creados desde un JCL, sino que se hacen mediante comandos AMS, y como se sale del objetivo de este artículo, no se trata este punto.

### 2.2.- Definición de ficheros GDG (Generation Data Group)

Estos son ficheros secuenciales normales que sólo tienen de especial el nombre ya que bajo un mismo nombre se pueden referenciar varios ficheros.

```
//XXXXXXXXDG JOB (12,345),'prueba',NOTIFY=XXXXXX,CLASS=x,MSGCLASS=u
/** ***** **
//PASO1      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN      DD *
      DEFINE GDG (NAME(JMPDES.EJEMPLO.GDG) LIMIT(5) NOEMPTY SCR)
/*
```

Figura 1: Esqueleto para crear un GDG

La definición de un fichero GDG consta de dos pasos: En el primero se crea lo que es el nombre del fichero en el catálogo. La **figura 1** muestra un paso de JCL para realizar este cometido. Como puede verse en él, no se especifican características, como ocurre cuando se crea cualquier fichero, ya que solo es una entrada en el catálogo de usuario, no en el VTOC pues todavía no existe físicamente.

Una vez creado el nombre, cuando se desee añadir una versión a ese fichero, se realizará lo siguiente:

- Añadir el sufijo (+1) al nombre
- Especificar las características de este fichero secuencial, que no tienen por que coincidir con las otras versiones,
- Poner como disposición inicial NEW.

En cambio, cuando se desee leer una versión habrá que especificar como nombre, el nombre del GDG y el sufijo (0) para la última versión, (-1) para la penúltima, etc. ; y como disposición inicial hay que especificar SHR (Shared: compartido)

(Ver *SOLO PROGRAMADORES num. 20*)

El formato del comando que define un GDG es el siguiente:

```
DEFINE GDG (-
      NAME(nombre_gdg)-
      LIMIT(num_ver) -
      EMPTY/NOEMPTY -
      SCRATCH/NOSCRATCH)
```

donde

**NAME:** Sirve para especificar el nombre del fichero GDG a definir

**nombre\_gdg** : representa el DSN del fichero.

**LIMIT**: Sirve para especificar el numero de versiones que se desea mantener.

**num\_ver** : numero de versiones de 1 a 155.

**EMPTY** : Opcional. Especifica que se descataloguen todos los nombres de fichero cuya versión supere la especificada en LIMIT.

**NOEMPTY** : Sirve para especificar que según se vayan generando versiones nuevas, se vayan borrando las mas antiguas, de forma que solo estén activas las ultimas LIMIT, tal y como muestra la **figura 2**, pudiendo ver el listado de la entrada del catalogo (no tiene VOLUMEN) y las 5 versiones (6 a 10) qu se especificaron en la figura 1.

**SCRATCH** : Opcional. Especifica que se borre físicamente el fichero cuando se descatalogue.

DSLISL - DATA SETS BEGINNING WITH JMPDES.E*.G* ----- ROW 1 OF 7			
COMMAND ==>>		SCROLL ==>> CSR	
COMMAND	NAME	MESSAGE	VOLUME
-----			
	JMPDES.EJEMPLO.GDG		??????
	JMPDES.EJEMPLO.GDG.G0006V00		X1DES91
	JMPDES.EJEMPLO.GDG.G0007V00		X1DES91
	JMPDES.EJEMPLO.GDG.G0008V00		X1DES91
	JMPDES.EJEMPLO.GDG.G0009V00		X1DES91
	JMPDES.EJEMPLO.GDG.G0010V00		X1DES91
***** END OF DATA SET LIST *****			

Figura 2: Nombres de las versiones 6 a 10 de un GDG

### 2.3.- Listar entradas del catalogo

El comando LISCAT permite listar entradas del catálogo. Su formato es el siguiente:

**LISCAT CATALOG (nom\_catalogo).**

### 2.4.- Copiar ficheros

El comando **REPRO** permite realizar las siguientes copias de ficheros:

- Copiar ficheros secuenciales a ficheros VSAM
  - Copiar ficheros VSAM a ficheros VSAM
  - Copiar ficheros VSAM a ficheros secuenciales no-VSAM
  - Copiar ficheros ISAM (Indexed Sequential Access Method) a ficheros VSAM.
- el esqueleto de un paso que realice esta función, sería:

```
//COPIAR EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//ENTRADA DD DSN=nombre_fichero_entrada...
//SALIDA DD DSN=nombre_fichero_salida ...
//SYSIN DD *
  REPRO -
  INFILE(ENTRADA) OUTFILE(SALIDA) COUNT(nnn)
/*
```

Donde **ENTRADA** y **SALIDA** son las Ddnames que tienen asociados los nombres de los ficheros de entrada y de salida; y

**nnn** representa el numero de registros que se desean copiar. en el caso de no especificar el parámetro **COUNT** se entiende que es todo el fichero.

### 2.5.- Imprimir ficheros

El comando **PRINT** permite imprimir tanto ficheros VSAM como ficheros NO-VSAM

El esqueleto de un paso que realice esta función es el siguiente:

```
//COPIAR EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//ENTRADA DD DSN=nombre_fichero_entrada...
//SYSIN DD *
  PRINT -
  INFILE(ENTRADA) formato
/*
```

Donde **ENTRADA** representa la Ddname que tiene asociado el nombre del fichero a imprimir; y

**formato** puede tomar uno de los siguientes valores:

- **CHAR** : imprime los registros en formato carácter.
- **HEX** : Imprime los registros en formato HEXADECIMAL.
- **DUMP** : Imprime los registro en ambos formatos.

## 2.6.- Borrar ficheros

El comando **DELETE** permite borrar tanto ficheros VSAM como ficheros NO-VSAM

El esqueleto de un paso que realice este comando es el siguiente:

```
//COPIAR EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//FICHERO DD DSN=nombre_fichero...
//SYSIN DD *
  DELETE nom_fichero -
    FILE(FICHERO) -
    PURGE          -
    ERASE          -
    SCRATCH        -
    CATALOG(nom_catalogo)
/*
```

Este comando admite que el fichero a borrar se especifique directamente o bien a través del parámetro **FILE** especificando en el nombre de la Ddname que tiene asociado.

El resto de parámetros es opcional, significando lo siguiente:

- PURGE** : Borra la entrada mirando la fecha en que expira el fichero.
- ERASE** : Borrar el fichero y sobrescribe ceros binarios.
- SCRATCH** : Se omite para ficheros VSAM.
- CATALOG** : Nombre del catalogo VSAM del que se desea borrar el ítem.

## 2.7.- SET MAXCC=nn

Este comando especifica que con independencia de cual haya sido el resultado de la ejecución anterior, devuelva como código de retorno el valor **nn** especificado.

### 3.- IKJEFT01:

El tema que se esta tratando es lo suficientemente amplio como para incluir en él a **IKJEFT01**, aunque al igual que en el punto anterior, el programa que pone título a este apartado, no sea realmente una utilidad, sino toda una facility o conjunto de programas que favorecen el desarrollo, pues es el propio **TSO**.

Por otra parte, como el autor entiende que este programa se merece algo mas que un apartado, solo va a comentar las Ddnames necesarias para ejecutar este programa en batch, lo cual permitirá:

- ejecutar comandos de TSO
- submitir JCLs,
- Enviar mensajes a otros usuarios
- etc.

**SYSTSPRT** Esta Ddname se corresponde con la **SYSPRINT** de las utilidades vistas anteriormente, y define el destino al que se dirigirán los mensajes generados por el programa.

**SYSTSIN**: Se corresponde con la **SYSDIN** y contiene los comandos que serán ejecutados por el programa. En el ejemplo de la figura 3, el comando **PRINTDS**.

```
***** ***** TOP OF DATA *****
000001 //XXXXXX6V JOB (30,000),'SOTANO-1 SALA-2',CLASS=T,MSGCLASS=X,
000002 //          NOTIFY=XXXXXX,MSGLEVEL=(0,0)
000003 //V158      OUTPUT CHARS=GTE5,PAGEDEF=388STD,FORMDEF=STDA
000004 //*=====
000005 //*          IMPRIMIR TEMPORAL CON MARGEN
000006 //*=====
000007 //*
000008 //PASOASEC EXEC PGM=IKJEFT01
000009 //*
000010 //SYSTSPRT DD SYSOUT=*
000011 //SYSTSIN DD *
000012 PR DATASET('JMPDES.SALIDA') +
000013 COLUMNS(1:80) PAGELEN(80) LMARGIN(10) ALL +
000014 DEST(LOCAL) OUTDES(V158)
000015 /*
```

Figura 3: Ejemplo de jcl para ejecutar el comando de TSO: PRINTDS

#### 4.- PRINTDS o PR:

Realmente, no se puede decir que se ajuste exactamente a la definición de utilidad dada en el primer artículo ya que es un comando de **TSO**, pero se incluye por no ser muy conocido en las instalaciones a nivel de desarrollador, y además, ser uno de los más útiles de TSO, pues, como su nombre indica, hace un **PRINT** de un data set (**DS**) o fichero, ya sea éste secuencial o particionado.

Entre sus características, cabe destacar las siguientes:

- Escribe cabeceras en cada página
- Se puede establecer margen superior(**TMARGIN**), Izquierdo(**LMARGIN**), espaciado,
- Clase (**CLASS**) e impresora de salida (**DEST**)
- o, se puede direccionar la salida a un fichero (**TODATASET**)
- en el caso de ficheros particionados, genera un índice al final del listado;
- se puede listar solo el directorio de un fichero particionado.

Para conocer la sintaxis y todas las opciones aceptadas por este potente comando, basta ejecutar : **TSO HELP PRINTDS**

Para ejecutar on-line desde TSO, solo hay que especificar desde la línea de comando: **TSO PRINTDS DSNNAME('JMPDES.JMP.SALIDA') DEST(RMT101)**

Para que el lector pueda tener un ejemplo completo del uso de este comando, la **figura 3** muestra el paso de un JCL que ejecuta el programa **IKJEFT01**, es decir TSO, y en la entrada correspondiente a los comandos a ejecutar, incluye el comando **PRINTDS** y la serie de parámetros necesarios para su ejecución.

Es de resaltar que cada línea especifica un comando, pero si todos los parámetros de un comando no caben en una línea, estas se pueden unir especificando el carácter '+' al final de la misma.

## 5 .- SYS1.HELP

Para cerrar el tema, se incluye la mención a esta librería propia del sistema, (tiene por primer cualificador : **SYS1**) ya que en ella se puede encontrar la sintaxis de muchos comandos y programas del sistema, así como la explicación de cada uno de los parámetros admitidos por el comando, y documentación de los mensajes de error que generan.

Como ejemplo de lo dicho en el párrafo anterior, se aconseja listar el miembro **IKJEFY55**, que muestra todo el **HELP** asociado al comando **PRINTDS** visto en el apartado anterior.

Pero, como a veces se desconoce el nombre del comando a ejecutar, no se puede pedir al sistema que nos muestre la ayuda asociada. Por esta razón se recomienda el uso del programa **ISRSUPC**, propio del sistema, con los parámetros y Ddnames que se muestran en la **figura 4**, el cual permite buscar una palabra, por ejemplo "MARGIN" entre los miembros de una librería, obteniendo de esta forma la relación de comandos que usan dicha palabra bien en su sintaxis, bien en el texto de ayuda que acompaña a dicho comando.

```
00010 //XXXXXXST JOB (12,345),'scan tso',CLASS=T,MSGCLASS=X,
00020 //          NOTIFY=XXXXXX,MSGLEVEL=(0,0)
00030 //*****
00040 //*          BUSCAR CADENA EN DIRECTORIO
00050 //*****
00060 //BUSCAR EXEC PGM=ISRSUPC,PARM=(SRCHCMP,'ANYC')
00070 //NEWDD DD DISP=SHR,DSN=sys1.help
00080 //OUTDD DD SYSOUT=*
00090 //SYSIN DD *
00100 SRCHFOR 'printds'
00110 /*
```

Figura 4: Ejemplo de jcl para  
buscar la cadena PRINTDS en la librería SYS1.HELP

## 6.- Utilidad

La utilidad que acompaña este mes al artículo retoma el JCL que acompañaba al primer artículo de este tema, para mostrar el resultado que se obtiene al ejecutar una utilidad desarrollada por el autor de este artículo y que tiene por finalidad listar los **objetos natural** cuyo nombre figura en un fichero que sirve de entrada y que se representa en dicho esqueleto con la variable **&ENTRADA**.

Puesto que el tema que nos ocupa no es NATURAL, no se comentan los programas invocados para ese entorno. Solo se acompaña dicho JCL para que se pueda tener un ejemplo real del uso de las distintas utilidades tratadas en este tema.

Uno de los resultados obtenidos con la ejecución de este JCL es el árbol jerárquico de objetos que se muestra en la **figura 5** correspondiente al Browser de NATURAL que se distribuyó con los artículos del tema anterior : Natural.

T	REF	LIN	DETALLE	DDM
		1		
		2	LISPGMOP . . . . .	6
		3		
C		4	+----> LISPGMER	
		5		
P		6	+----> MENU	
		7		
N		8	+----> LISPGM0H	
		9		
P	2	10	+----> LISPGMOP (*)	
		11		
P		12	+----> LISPGM1P . . . . .	6
		13		
C	4	24	+----> LISPGMER (*)	
		25		

**Figura 5: Arbol de objetos NATURAL correspondientes al Browser cuyos fuentes se acompañaron con el tema: OBJETOS NATURAL.**

Por último, desde estas líneas se lanza un mensaje a nuestros lectores para que aquellos que quieran ver publicadas las utilidades desarrolladas por ellos mismos, las remitan junto con una breve descripción de la misma a la redacción de la revista, o bien a la dirección del autor: **<jmpeco@luznet.es>**. También pueden remitir aquellas direcciones y foros de INTERNET que consideren interesantes y relacionados con el tema de **Grandes Sistemas**.

**ANEXO.- Bibliografía:**

**MVS manual para programadores**

ISBN : 84-481-0092-1

Editorial : McGraw-Hill

**EXPERT MVS/ESA JCL a guide to advanced Techniques**

ISBN: 0-07-009820-4

Editorial : McGraw-Hill

**SYSTEM/370 Job Control Language**

Autor: Gary DeWARD BROWN

Editorial: JOHN WILEY&SONS

**TSO para desarrolladores (JCL/CLIST/ISPF)**

Num.RPI: M-33933

Inedito

Autor: José María Peco (En el disco que acompaña a la revista se encuentra el fichero **LIBRO.DOC** que contiene el índice y una presentación del mismo junto con un browser freeware de Microsoft para documentos WORD).