

¿Como navegar por MVS conociendo las librerias alocadas ?

o

¿Como capturar la salida de un comando no direccionable a fichero en MVS ? (2<sup>oa</sup> parte)

En el articulo del mes pasado se trataba de capturar la salida de un comando no direccionable a un fichero, cosa que se realizo haciendo uso del comando de TSO que lista las librerias que se encuentran alocadas o asignadas al entorno del usuario: LISTALL o abreviadamente LISTA.

El presente articulo pretende ser la continuacion de aquel, y tiene por objetivo tratar el fichero obtenido en el mismo, para asi poder navegar por el sistema mediante paneles de seleccion. Para llevar a cabo este objetivo, se va a hacer uso del fichero obtenido al ejecutar el comando LISTA, pero con la opcion STATUS, lo cual permite obtener no solo las librerias asignadas al entorno del usuario, o alocadas, sino tambien el nombre logico que tienen asociado, y su disposicion.

```
----- TSO COMMAND PROCESSOR -----
ENTER TSO COMMAND, CLIST, OR REXX EXEC BELOW:

==> lista status

--DDNAME---DISP--
DESJMP.ISPF.ISPPROF
  ISPPROF  KEEP
TERMFIL  SYSPRINT
TERMFIL  SYSOUT
TERMFIL  SYSTEM
TERMFIL  SYSTSPRT
TERMFIL  SYSIN
TERMFIL  SYSTSIN
DESJMP.SPFLOG1.LIST
  ISP14421 CATLG
DESJMP.JMP.PANEL
  ISP14422 KEEP,KEEP
DESJMP.JMP.ESKELETO
  ISP14423 KEEP,KEEP
DESJMP.JMP.MENSAJES
  ISP14424 KEEP,KEEP
DESJMP.INT.TABLA
  ISP14425 KEEP,KEEP
DESJMP.JMP.TMP
  SALIDA  KEEP
*DESJMP.JMP.JCL
  SYS00061 KEEP
***
```

Fig. 1: Lista proporcionada por el sistema al ejecutar el comando : LISTA STATUS

```
----- Lista de librerias ----- ROW 1 TO 16 OF 94
OPCION====> SCROLL ==> CSR

Opciones : B Browse      <<< LIBRERIAS ALOCADAS >>>_____
           E Edit
           M Miembros

m          Librerias          Observaciones
-----
DESJMP.ISPF.ISPPROF          ISPPROF
DESJMP.JMP.ESKELETO          ISP14423
DESJMP.JMP.MENSAJES          ISP14424
DESJMP.JMP.PANEL             ISP14422
DESJMP.JMP.TMP               SALIDA
DESJMP.SPFLOG1.LIST          ISP14421 CATLG
TERMFIL                      SYSPRINT
***** BOTTOM OF DATA *****
```

Fig. 2: Pantalla que permite visualizar o editar las librerias listadas en la figura anterior.



```

IF      &CONTA > &ULTIMO THEN GOTO CERRAR          00007100
SET     &LINEA1 = &&NOMBRE&CONTA                    00007200
SET     &LINEA1 = &STR(&LINEA1 &BLANCOS)           00007300
SET     &LINEA1 = &STR(&SUBSTR(1:44,&STR(&LINEA1))) 00007400
SET     &LINEA2 = &STR( )                          00007500
IF &STR(&SUBSTR(1:8,&STR(&LINEA1))) = TERMFIL +      00007600
  THEN DO                                         00007700
/*      SET &LINEA1(= &STR(&SUBSTR(01:10,&STR(&LINEA1))) */ 00007800
      SET &LINEA2 = &STR(&SUBSTR(09:23,&STR(&LINEA1))) 00007900
      GOTO SEGUIR                                  00008000
      END                                          00008100
SET     &CONTA = &CONTA + 1                       00008200
IF      &CONTA > &ULTIMO THEN GOTO CERRAR          00008300
SET     &LINEA2 = &&NOMBRE&CONTA                    00008400
SET     &LINEA2 = &STR(&LINEA2 &BLANCOS)           00008500
SET     &LINEA2 = &STR(&SUBSTR(1:44,&STR(&LINEA2))) 00008600
/*      SET ELEMENTO = &STR(&SUBSTR(01:44,&LINEA1)) */ 00008700
/*      SET COMENTA = &STR(&SUBSTR(01:15,&LINEA2)) */ 00008800
SEGUIR: SET &SALIDA = &STR(&LINEA1&LINEA2)         00008900
        PUTFILE SALIDA                             00009000
        IF &TEST = TEST THEN WRITE &SALIDA         00009100
        GOTO GRABAR                                 00009200
CERRAR: CLOSFIL SALIDA                             00009300
        FREE FILE(SALIDA)                          00009400

```

Al llegar a este punto, se dispone de un fichero secuencial con las librerías asignadas al entorno, y, escrito en el mismo registro, el nombre lógico que tienen asignado, y la disposición.

```

/*      * / 00009500
/* ===== DISPLAYA LA TABLA DE ELEMENTOS ===== * / 00009600
/*      * / 00009700
CONTROL MSG                                       00009800
SET EOF = NO                                     00009900
SET RETCOD = 0                                  00010100

```

Esta primera parte, lo que hace es inicializar las variables EOF de (End-Of-File) para saber cuando se detecta el fin de fichero. RETCOD de (Return Code) para memorizar el código de retorno devuelto por los servicios de ISPF que se invoquen.

```

ERROR +                                         00010300
DO                                             00010400
  SET RETCOD = &LASTCC                          00010500
  IF &RETCOD = 8 THEN +                         00010600
    DO                                         00010700
      IF &PASO = EXAMEN THEN +                 00010800
        DO                                     00010900
          SET EXISTE = NO                       00011000
          RETURN                                00011100
        END                                     00011200
      WRITENR YA EXISTE LA TABLA ENTORNO0. NO PUEDO CREAMLA 00011300
      READ SEGUIR                              00011400
      EXIT (RETCOD)                            00011500
    END                                         00011600
  IF &RETCOD = 400 THEN +                       00011700
    DO                                         00011800
      SET EOF = SI                             00011900
      RETURN                                    00012000
    END                                         00012100
  ELSE +                                       00012200
    EXIT (RETCOD)                              00012300
  END                                          00012400

```

10500 memoriza el código de retorno que ha generado la cesión del control a esta rutina. Dicho código se encuentra contenido en la variable del sistema &LASTCC.

10600 Examina si el código de retorno es 8, y si es así ejecuta las instrucciones contenidas entre las línea 10700-11600.

10800 Examina el contenido de una variable que se carga en el flujo de ejecución normal, y que sirve para detectar el punto desde el que ha bifurcado a esta rutina ya que el código de retorno 8 es devuelto por más de una función. Concretamente esta examinando si la variable PASO contiene la cadena EXAMEN, valor que se carga en la línea 13700.

- 11000 En el caso de que contenga dicho valor, quiere decir que no existe una fila con la clave que se esta buscando, cosa que es correcta ya que se quiere dar de alta una entrada en una tabla indexada, y para ello necesitamos que no exista. Por esta razon es por lo que se carga la variable EXISTE con el valor NO.
- 11100 Retorna al punto desde el que se bifurco a esta rutina.
- 11300 Escribe el mensaje de que se quiere crear una tabla que ya existe.
- 11400 Se interrumpe la ejecucion de la clist hasta que el usuario pulse INTRO. Esto se consigue indicando al sistema que lea desde el terminal el valor de la variable SEGUIR, la cual no sirve mas que para interrumpir la ejecucion de la clist, a la espera de que se pulse INTRO.
- 11500 Se sale de la clist devolviendo el codigo de retorno 99
- 11700 En el caso de que el codigo que provoco esta bifurcacion incondicional no fuera el 8 (linea 10600) examina si es el codigo 400, generado cuando en la lectura de un fichero se ha detectado el fin de fichero.
- 11900 si se ha detectado esta condicion, se carga la variable de usuario EOF con el valor SI y se regresa con RETURN al punto desde el que se bifurco a esta rutina de errores.
- 12300 En el caso de no ser ninguno de estos errores, sale de la clist devolviendo el codigo de retorno que ha provocado el error.

Los dos siguientes grupos de sentencias constituyen el procedimiento para el que se ha escrito la rutina de errores previa. Solo quiero comentar que una tabla ISPF no es mas que una tabla o fichero indexado de doble entrada; tiene tantas filas como registros, y cada registro tiene tantos campos como filas tiene definida la tabla.

ALLOC FI(ENTRADA) DA(&FICHERO) SHR REUSE	00012500
ISPEXEC TBCREATE ENTORNO0 NOWRITE NAMES(ELEMENTO,COMENTA)	00012600
OPENFILE ENTRADA	00012700
/*	00012800
GETFILE ENTRADA	00012900
/* SET LIBRERIA = &STR(&SUBSTR(1:45,&ENTRADA))	00013000
/*	00013100
GETFILE ENTRADA	00013200

- 12500 Aloca el dataset cuyo nombre se encuentra contenido en la variable FICHERO para ser usado de entrada con disposicion SHARED, y le asigna el nombre logico de ENTRADA.
- 12600 Llama al servicio de ISPF : TBCREATE para crear una tabla ISPF temporal (NOWRITE) con los campos de nombre ELEMENTO y COMENTA.  
ISPF, por haber invocado una de sus funciones, nos devuelve siempre un codigo de retorno.  
Si este codigo de retorno no es 0 salta a la rutina de errores que se ha comentado. Como lo normal es que en este caso sea 0, pues no existe la tabla ya que es en memoria, sigue en secuencia.  
Si ya existiera la tabla, devuelve el codigo de retorno 8. de ahi el mensaje de la linea 11300.
- 12700 Abre el fichero ENTRADA en modo INPUT.
- 12900 Lee el primer registro, que no interesa ya que solo contiene un texto de cabecera fijo (linea 4900 y 5000).
- 11000 Lee el siguiente registro, el cual ya contiene informacion valida.

Por lo tanto, con un registro valido leido, se entra en la repetitiva siguiente:

DO WHILE &EOF = NO	00013300
SET ELEMENTO = &STR(&SUBSTR(01:44,&ENTRADA))	00013400
SET COMENTA = &STR(&SUBSTR(45:60,&ENTRADA))	00013500
/*	00013600
SET PASO = EXAMEN	00013700
SET EXISTE = SI	00013800
ISPEXEC TBEXIST ENTORNO0	00013900
IF &EXISTE = NO THEN ISPEXEC TBADD ENTORNO0 SAVE(ELEMENTO,COMENTA)	00014000
GETFILE ENTRADA	00014100
END	00014200

CLOSFILE ENTRADA	00014300
FREE FI(ENTRADA)	00014400
ERROR OFF	00014500

- 13300 Define un estructura repetitiva que se ejecutara mientras el contenido de EOF sea NO
- 13400 Carga en la variable ELEMENTO las posiciones de 1 a 44 de la variable ENTRADA (buffer que contiene el registro leído)
- 13500 Carga en la variable COMENTA las posiciones 45 a 60 de ENTRADA.
- 13700 Carga en la variable PASO la cadena EXAMEN. Esta variable es la que se examina en la rutina de ERROR (10800).
- 13800 Carga en la variable EXISTE la cadena SI
- 13900 Llama a la funcion de ISPF : TBEXIST para saber si existe una fila (registro) en la tabla ispf de nombre ENTORNO0.  
Llegados a este punto se puede producir una situacion de excepcion, pues puede existir o no una fila con el valor contenido en la variable ELEMENTO, dependiendo de que esten duplicados los nombres de las librerias, con lo que esta funcion puede retornar:  
- el codigo 0 si existe la fila, y sigue en secuencia, o  
- el codigo 8 si no existe la fila, y por tanto bifurca a la rutina de error. Una vez en esta rutina, pregunta por el codigo, y como sabe de donde viene por el contenido de la variable PASO, carga el valor NO en la variable EXISTE y retorna a la siguiente sentencia a la que provocø su bifurcacion. Es decir, retorna a la sentencia 14000
- 14000 Pregunta por el contenido de la variable EXISTE.  
En el caso de que el contenido sea NO, es decir no existe una fila con la entrada dada, llama a la funcion de ISPF: TBADD para añadir la fila a la tabla, con los campos ELEMENTO Y COMENTA.
- 14100 Lee un nuevo registro del fichero ENTRADA
- 14200 Cierra el bucle de la repetitiva DO-WHILE. Este bucle se repetira hasta que al leer un nuevo registro se detecte el fin de fichero, con lo que se bifurcara de nuevo a la rutina de error, y al comprobar que el codigo de retorno es 400, cargar en la variable EOF el valor SI, por lo que al retornar de nuevo a este punto, e ir a la cabecera de la repetitiva, comprobara que el contenido de la variable EOF no es NO, y saltara a la linea 12100.
- 14300 Cierra el fichero asociado al nombre logico ENTRADA
- 14400 Libera para el sistema el fichero fisico asociado al nombre logico ENTRADA.
- 14500 Desactiva la rutina de error.

Asi pues, al llegar a este punto ya se dispone de una tabla ISPF cargada con los datos que interesan.

Las siguientes sentencias presentan la parte de la clist que muestra la tabla en un panel, y permite tratar la informacion contenida en cada una de sus lineas.

ISPEXEC TBSCAN ENTORNO0 ARGLIST(ELEMENTO) NEXT	00014600
ISPEXEC TBSORT ENTORNO0 FIELDS(ELEMENTO,C,A)	00014700
ISPEXEC LIBDEF ISPLLIB DATASET ID(' &SYSUID..JMP.PANEL')	00014750

- 14600 establece como argumento de busqueda para un SCAN , el campo ELEMENTO, y la busqueda se realizara hacia delante.(NEXT)
- 14700 Clasifica la tabla por el campo ELEMENTO, en ascendente, y su contenido, aunque sea numerico, sera tratado como caracteres alfanumericos.
- 14750 Define como libreria de paneles la libreria especificada.

El siguiente fragmento contiene una repetitiva a fin de tratar todos los elementos de la tabla.

SET &SIGUE = SI	00014800
DO WHILE &SIGUE = SI	00014900
ISPEXEC TBTOP ENTORNO0	00015000
ISPEXEC TBDISPL ENTORNO0 PANEL(ENTORNO0)	00015100
IF &LASTCC = 8 THEN -	00015200
DO	00015300
ISPEXEC TBCLOSE ENTORNO0	00015400
EXIT (99)	00015500
END	00015600

```

SET &FICHERO = &ELEMENTO                                00015900
IF &MARCA = B THEN -                                    00016000
DO                                                       00016100
    ISPEXEC BROWSE DATASET('&FICHERO')                 00016200
    SET &COMENTA = 'VISUALIZADA'                       00016300
END                                                       00016400
IF &MARCA = E THEN -                                    00016500
DO                                                       00016600
    ISPEXEC EDIT DATASET('&FICHERO')                 00016700
    SET &COMENTA = 'EDITADA'                          00016800
END                                                       00016900
IF &MARCA = M THEN -                                    00017000
DO                                                       00017100
    EXEC '&SYSUID..OPER.OPERCLIB(ENTORNOM)', '&FICHERO' 00017200
END                                                       00017300
SET &MARCA = &STR()                                    00017400
END                                                       00017500

```

- 14800 Inicializa la variable SIGUE con el valor SI
- 14900 Inicia un bucle que se ejecutara mientras la variable SIGUE contenga el valor SI
- 15000 posiciona el puntero en el la primera fila de la tabla clasificada.
- 15100 LLama a la funcion de ISPF para presentar en un panel las filas de la TABLA. ISPF ya se encarga de la paginacion, numero de filas por pagina, etc.
- 15200 Al no haber una rutina de error a la que bifurcar, despues de cada llamada se debe preguntar por el codigo de retorno que devuelve ISPF. Esto es lo que se hace en esta linea.  
En el caso de que el codigo sea 8 , es decir no hay filas, cierra la tabla (15400), y sale de la clist.
- 15900 Carga la variable FICHERO con el contenido del campo ELEMENTO de la fila seleccionada.
- 16200 llama a la funcion de ISPF : BROWSE pasandole como argumento el contenido de la variable FICHERO.
- 16300 Mueve el texto VISUALIZADA en la variable COMENTA de la fila seleccionada.
- 16700 llama a la funcion de ISPF : EDIT pasandole como argumento el contenido de la variable FICHERO.
- 16800 Mueve el texto EDITADA en la variable COMENTA de la fila seleccionada.
- 17200 Llama a la clist ENTORNOM, pasandole como argumento el contenido de la variable FICHERO con el fin de presentar los miembros de una libreria, y tratarles de forma parecida a como se hace con este panel.
- 17400 mueve blancos a la variable MARCA
- 17500 Cierra el bucle de la repetitiva.

Por ultimo, se presenta la definicion del panel ENTORNO0, que para que pueda ser tratado, debe ser un miembro de la libreria de paneles del usuario: &SYSUID..JMP.PANEL (linea 14750), donde &SYSUID es una variable del sistema que contiene el Identificador del usuario de TSO.

```

)ATTR default($+_
? TYPE(INPUT) INTENS(low) CAPS(off) JUST(LEFT)
# TYPE(OUTPUT) INTENS(high) CAPS(ON) JUST(LEFT) pad(_)
)BODY
$----- Lista de librerias -----
$ OPCION====>_Zcmd +SCROLL ==> _scin+
$
+ Opciones :$B+Browse #cabecera +
+ $E+Edit
+ $M+Miembros
+ m Librerias Observaciones
+ -----
)MODEL ROWS(ALL)
+ _z?elemento +?comenta +
)INIT
&SCIN= Csr
.zvars = marca
&marca = ' '
)END

```