

TSO: COMANDI

F. Benedetti

CNUCE-211

5 ottobre 1987

**COMANDI
TSO**

INTRODUZIONE

Il presente manuale vuole essere una descrizione introduttiva ai comandi TSO e come tale fornisce informazioni riguardanti quelli più utilizzati.

Alcuni comandi, come alcuni sottocomandi e operandi di quelli presenti non appaiono in questo manuale e per essi si rimanda a quello ufficiale dell'IBM:

TSO Extentions Command Language Refence.

A cura di
F. BENEDETTI

Un comando **TSO** consiste in un nome di comando seguito, di solito, da 1 o più operandi. Gli operandi provvedono affinché le informazioni richieste per il comando eseguano l'operazione richiesta. Esistono due tipi di operandi per i comandi: posizionali e parole chiavi.

Gli operandi **Positional** seguono il comando in una sequenza predefinita. Quando si vuol dare un operando positional consistente in una lista di nomi o valori, la lista deve essere inclusa tra parentesi che si devono bilanciare.

Gli operandi **KEYWORD** sono nomi specifici o simboli che hanno un particolare significato per il sistema. Si possono includere operandi keyword in qualsiasi ordine dopo gli operandi positional.

I commenti possono essere aggiunti dovunque dopo almeno un blank, basta semplicemente delimitarli con /* e */.

Per dividere i comandi dal primo operando occorrono 1 o più blanks, per separare gli operandi si possono usare i blanks o una virgola.

Se occorre continuare nella linea successiva si può usare o il segno + o il segno - come ultimo carattere della riga su cui stiamo lavorando.

Alcuni comandi come **EDIT**, **OUTPUT** hanno a disposizione anche alcuni sottocomandi che possono essere dati quando il comando è diviso in operazioni individuali. E' possibile continuare a dare i sottocomandi finché non viene dato il sottocomando **END**.

I data-sets in **TSO** sono composti da:

- 1) lo user-prefix,
- 2) un nome supplementare,
- 3) un qualificatore descrittore.

I qualificatori descrittivi dovrebbero essere sempre uno dei seguenti:

ASM	per i data-sets di assembler
CLIST	per i data-sets contenenti i comandi TSO
CNTL	per i data-sets contenenti JCL e SYSIN per il SUBMIT
COBOL	per i data-sets di cobol
DATA	per i data-sets contenenti un testo in maiuscolo
FORT	per i data-sets in fortran
LINKLIST	listing di output del linkage editor
LIST	listing normale
LOAD	un modulo di load
LOADLIST	listing di output del loader
OBJ	modulo oggetto
OUTLIST	listing di output dal comando output
PLI	data-sets di pl/1
TESTLIST	listing di output dal comando test
TEXT	data-sets contenenti un testo maiuscolo e minuscolo
VS BASIC	data-sets di vsbasic

L'utilità dei qualificatori descrittivi consiste nel fatto di, se il nome del data-set è conforme alle convenzioni, poter specificare soltanto lo user e il nome supplementare quando si vuol riferire un data-set.

Addirittura con alcuni comandi (come LINK per es.) basta specificare il nome supplementare (seguito dal nome del membro se si tratta di un partitioned) per richiamare un data-set.

I comandi **TSO** si possono suddividere nelle seguenti fasce:

A) comandi per controllare la sessione a terminale:

HELP, LOGOFF, LOGON, RECEIVE, SEND, TRANSMIT.

B) comandi per la gestione dei dati:

ALLOCATE, ATTRIB, DELETE, EDIT (con i suoi sottocomandi)
FREE, LISTALC, LISTBC, LISTCAT, LISTDS, RENAME.

C) comandi per l'esecuzione e il debug dei programmi:

CALL, LINK, LOADGO, RUN.

D) comandi per la gestione dei programmi batch:

CANCEL, OUTPUT, SUBMIT, STATUS.

**COMANDI
PER
CONTROLLARE
LA
SESSIONE
A
TERMINALE**

HELP

HELP [(sub)command-name]
H [FUNCTION]
[SINTAX]
[OPERANDS]
[ALL]

Il comando **HELP** viene utilizzato per ottenere le informazioni relative alle funzioni, alle sintassi, agli operandi di comandi e sottocomandi e informazioni circa determinati messaggi. Dando il comando **HELP** si ottiene una lista di tutti i comandi **TSO** raggruppati per funzioni o i sottocomandi del comando che si sta utilizzando.

command-name o **subcommand-name** specifica il nome c comando o sottocomando di cui si vogliono le informazioni.

FUNCTION si vuol sapere qualcosa di più circa lo scopo e le operazioni del comando o sottocomando.

SINTAX si vuol conoscere qualcosa di più circa la sintassi richiesta per utilizzare il comando o il sottocomando.

OPERANDS si vuol vedere le spiegazioni degli operandi del comando o del sottocomando.

ALL si vuol conoscere tutte le informazioni relative al comando o al sottocomando.

LOGOFF

LOGOFF [DISCONNECT]
[HOLD]

Il comando **LOGOFF** viene utilizzato per terminare la propria sessione a terminale. Quando viene dato il comando, il sistema libera tutti i data-sets allocati. Se si vuole iniziare una nuova sessione utilizzando un'altra macchina **TSO** non occorre dare il comando **LOGOFF** basta battere il comando **LOGON** e automaticamente il sistema termina una sessione per iniziarne una nuova.

DISCONNECT specifica che la linea deve essere disconnessa dopo il comando. (default).

HOLD specifica che la linea non viene rilasciata dopo il comando.

LOGON

LOGON user-identity [/password]

Il comando **LOGON** viene usato per iniziare una sessione a terminale. Dopo il comando il **TSO** provvede a richiedere l'identificazione (**max 7 caratteri**), che è il nome della macchina **TSO**, e la password (**max 8 caratteri**).

user-identity specifica l'identificazione dell'utilizzatore (macchina **TSO**).

password la parola chiave per accedere alle risorse di quella user-identity.

RECEIVE

RECEIVE

[INDDNAME (ddname)]
[INFILE (ddname)]

[INDSNAME (dsn)]
[INDATASET (dsn)]

Il comando **RECEIVE** serve per ricevere un file trasmesso con comando **TRANSMIT** o proveniente da un altro nodo di rete e memorizzarlo nella sua forma originale.

INDDNAME o **INFILE** specifica l'uso di un file preallocato in cui ricevere il dato trasmesso.

INDSNAME o **INDATASET** specifica l'uso di un data-set sequenziale in cui ricevere il dato trasmesso.

SEND

SEND
SE

'text'

[USER (userid-list)]

[NOW]
[LOGON]

[OPERATOR (2)]

Il comando **SEND** viene utilizzato per inviare messaggi ad un altro utente o all'operatore. E' possibile inviare messaggi a uno o più utenti. Se il ricevente non è logon il sistema può ritenere il messaggio e presentarlo automaticamente quando il ricevente si ricollega.

'text' è il messaggio da inviare, deve essere racchiuso tra 2 apostrofi e non può essere più lungo di 115 caratteri inclusi i blanks.

USER (userid-list) specifica l'utente, o gli utenti, che deve ricevere il messaggio.

NOW si vuole che il messaggio venga spedito immediatamente.

LOGON il messaggio viene ritenuto dal data-set **SYS1.BROADCAST** e viene spedito immediatamente se il ricevente è logon altrimenti verrà spedito quando avrà fatto logon.

OPERATOR (2) specifica che il messaggio viene inviato alla master console dell'operatore.

TRANSMIT

TRANSMIT
XMIT

(addressee-list)
[DATASET (dsn)]
[DSNAME (dsn)]

[DDNAME (ddname)]
[FILE (ddname)]

[MEMBERS (memberlist)]

[SEQUENTIAL]

Il comando **TRANSMIT** viene utilizzato per inviare un messaggio, un data-set, o entrambi ad un altro utente che può essere locale o su un altro nodo di rete. Il comando converte i dati in un formato speciale che soltanto con il comando **RECEIVE** può essere riletto.

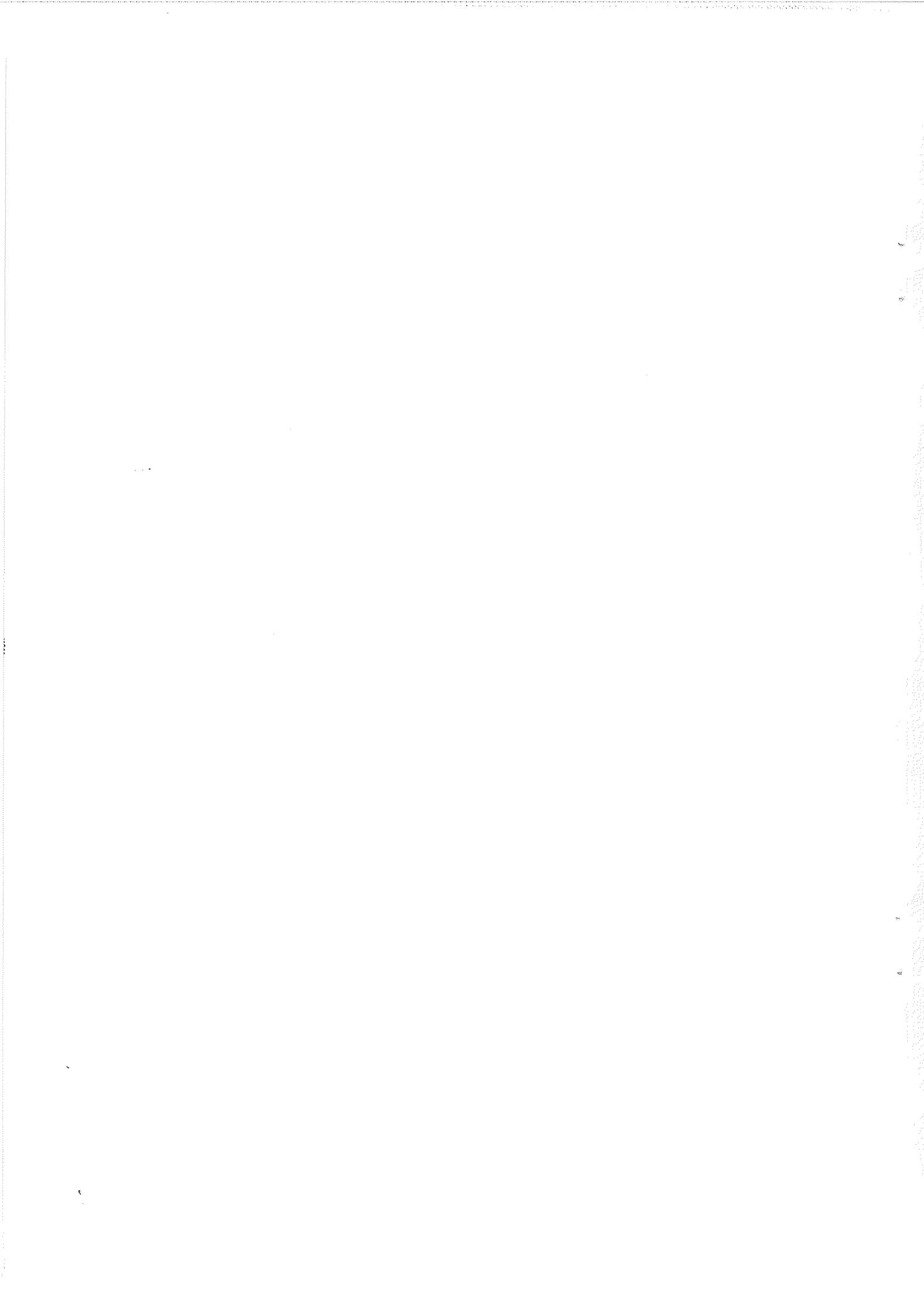
(addressee-list) specifica le informazioni relative ad identificare ricevente. Se è formato da un solo nome si possono omettere le parentesi.

DATASET o **DSNAME** (dsn) è il nome del data-set da trasmettere, deve essere un data-set su disco.

DDNAME o **FILE** (ddname) è il nome della DD di un file preallocato che deve essere trasmesso.

MEMBERS (memberlist) è la lista dei membri di un data-set partitioned che devono essere trasmessi.

SEQUENTIAL deve essere specificato per trasmettere un membro di un data-set partitioned a un utente **CMS**.



COMANDI PER LA GESTIONE DEI DATI

ALLOCATE

ALLOCATE
ALLOC

DATASET (*) [FILE (name)]
DSNAME(dsname-list) [DDNAME (name)]

[OLD]
[SHR]
[MOD]
[NEW]
[SYSOUT (class)]

[VOLUME (serial-list)]

[SPACE(quantity[,increment])
BLOCK (value)
TRACKS
CYLINDERS]

[UNIT (type)]

[LIKE (model-dsname)]

[KEEP]

[LRECL]

[RECFM]

[DSORG]

Il comando **ALLOCATE** viene utilizzato per allocare dinamicamente data-sets che i programmi richiederanno durante l'esecuzione.

DATASET o **DSNAME** specifica il nome del data-set che deve essere allocato. Se seguirà una lista di data-set names il comando allocherà e concatenerà dei data-sets non-Vsam.

Se viene messo un * al posto del nome del data-set vuol dire che si vuol avere il terminale allocato per l'input e l'output. Con * dobbiamo usare solo **FILE** o **DDNAME**, **BLOCK** o **BLKSIZE** gli altri operandi sono ignorati.

FILE o **DDNAME** seguito dal nome specifica il nome da associare al data-set. Questo nome corrisponde al nome della **DD** di un **JCL** e deve essere uguale alla **DDNAME** della **DCB** associata al data-set.

OLD indica che il data-set esiste e che se ne richiede l'uso esclusivo.

SHR indica che il data-set esiste ma non se ne richiede l'uso esclusivo.

MOD indica che vogliamo appendere in coda i dati.

NEW indica che deve essere creato un nuovo data-set.

SYSOUT (class) indica che dovrebbe trattarsi di un system output data-set.

VOLUME specifica il numero seriale di un disco su cui risiede il data-set. Per un data-set **NEW** occorre specificare il

SPACE	VOLUME altrimenti viene allocato su un volume residente su una unità specificata con il parametro UNIT o sul default stabilito dall'installazione. specifica l'ammontare dello spazio per allocare il data-set, vengono utilizzati i parametri del JCL e l'ammontare si ottiene come segue: BLOCK o BLKSIZE moltiplica il valore per la quantità dello SPACE e specifica la lunghezza media dei records; TRACKS la quantità dello SPACE è in numero di tracce; CYLINDERS la quantità dello SPACE è in numero di cilindri.
UNIT	specifica il tipo di unità su cui il data-set sarà allocato.
LIKE	specifica il nome di un data-set esistente i cui attributi devono essere utilizzati per un nuovo data-set che deve essere allocato.
KEEP	specifica che il data-set viene ritenuto dal sistema dopo che è stato liberato.
LRECL	specifica la più grande lunghezza del record logico.
RECFM	specifica il formato e la caratteristica del record dei data-sets, gli stessi della DCB del JCL .
DSORG	specifica l'organizzazione del data-set (vedi il JCL).

ATTRIB

ATTRIB
ATTR

attr-list-name

[BLKSIZE (blocksize)]

[LRECL (logical record length)]

[RECFM (A,B,D,F,M,T,U,V)]

[DSORG (valore)]

Si usa il comando **ATTRIB** per costruire una lista di attributi per un data-set non-Vsam che si intende allocare dinamicamente.

attr-list-name	specifica il nome della lista di attributi.
BLKSIZE	specifica la lunghezza del blocco del data-set, la lunghezza non eccederà i 32760 bytes.
LRECL	specifica la lunghezza, in bytes, del più grande record logico del data-set.
RECFM	specifica il formato e le caratteristiche del record del data-set. A il record contiene caratteri di controllo ASCII per la stampante; B i records sono bloccati; D i records sono a lunghezza variabile ASCII ; F i records sono a lunghezza fissa;
	M il record contiene il carattere di controllo del machine code;

DSORG

T il record può essere scritto in tracce di overflow se richiesto.

U i records sono a lunghezza indefinita;

V i records sono a lunghezza variabile.

specifica l'organizzazione come segue:

DA accesso diretto;

DAU accesso diretto non movibile;

PO organizzazione partitioned;

POU organizzazione partitioned non movibile;

PS organizzazione sequenziale;

PSU organizzazione sequenziale non movibile.

DELETE

DELETE
DEL

(entryname [/password] [.....])

[FILE (ddname)]

[PURGE]

[NOPURGE]

[SCRATCH]

[NOSCRATCH]

[NSCR]

Il comando **DELETE** viene utilizzato per cancellare uno o più data-sets o uno o più membri di un data-set partitioned. L'entry del catalogo viene rimossa solamente quando un intero data-set partitioned è cancellato.

entryname	è il nome delle entries nei cataloghi designati che devono essere cancellati. Se le entries sono più di una allora devono essere racchiuse tra parentesi. E' il primo parametro che deve seguire il comando DELETE .
FILE (ddname)	specifica il nome della DD che identifica il volume che contiene il data-set di cui dobbiamo fare il DELETE .
PURGE	specifica che si vuol cancellare il data-set anche se il periodo di ritenzione non è scaduto.
NO PURGE	si può cancellare solo se è scaduto il tempo di ritenzione. (default).
SCRATCH	il data-set viene cancellato. (default)
NO SCRATCH	il data-set non viene cancellato ma viene solo tolto dal catalogo.

EDIT

EDIT
E

data-set-name [/password]

[EMODE]

[IMODE]

[ASM]

[COBOL]

[CLIST]
[CNTL]
[TEXT]
[DATA]
[FORTGI]
[FORTH]
[PLI]
[PLIF]
[VSBASIC]

[SCAN]
[NOSCAN]

[NUM]
[NONUM]

[BLOCK (integer)]
[BLKSIZE (integer)]

[LINE (integer)]
[LRECL (integer)]

Il comando **EDIT** permette di creare, modificare, memorizzare, sottomettere, salvare e cancellare data-sets con organizzazione sequenziale o partitioned.

I data-sets possono contenere:

- programmi sorgenti di linguaggi come **PL/1**, **COBOL**, **FORTRAN** ecc.;
- dati da usare come input di un programma;
- text usati per informazioni di memoria o di recupero;
- comandi, sottocomandi, istruzioni **CLIST** e/o dati;
- istruzioni **JCL** per preparazione lavori.

Il comando sopporta soltanto data-sets che hanno uno dei seguenti formati:

- F;
- FB;
- V;
- VB.

data-set-name specifica il nome del data-set che si vuol editare o creare.

EMODE è il modo di entrata per i data-set **OLD**.

IMODE è il modo di entrata per i data-set **NEW**.

I vari tipi di data-sets nel comando **EDIT** sono:

ASM per l'assembler;
COBOL per il cobol;
CLIST per procedure di comandi;
CNTL per il **JCL**;
TEXT per testi da formattare;
DATA per files di dati;
FORTGI per il fortran-IV;
FORTH per il fortran-H esteso;
PLI per il pl/1;
PLIF per il pl/1 optimizing o checkout;
VSBASIC per il basic.

Il tipo va inserito se non fa già parte del nome del data-set.

SCAN	fa sì che ogni linea di dati che viene immessa deve essere controllata, istruzione per istruzione, per una sintassi corretta.
NOSCAN	non viene immesso il controllo automatico (default).
NUM	le linee dei records vengono numerate (default).
NONUM	le linee dei records non contengono i numeri di linea.
BLOCK o BLKSIZE	specifica la massima lunghezza, in bytes, per blocchi di records in un nuovo data-set.
LINE o LRECL	specifica la lunghezza, in bytes, di ogni record in un data-set nuovo.

Se gli ultimi due operandi sono omessi esistono dei valori di default per alcuni tipi:

TIPO	DSORG	LRECL	BLOCK	NUMERAZIONE
ASM	PS/PO	80=80	3120	ultime 8
CLIST	PS/PO	255≤255	3120	(F)ultime 8 (V)prime 8
CNTL	PS/PO	80=80	3120	ultime 8
COBOL	PS/PO	80=80	400	prime 6
DATA	PS/PO	80≤255	3120	ultime 8
FORTGI	PS/PO	255=80	400	ultime 8
FORTH	PS/PO	255=80	400	ultime 8
PLI	PS/PO	104≤100	400	come per CLIST
PLIF	PS/PO	80≤100	400	ultime 8
TEXT	PS/PO	255≤255	3120	come per CLIST
VSBASIC	PS/PO	255=80	3120	prime 5

L'**EDIT** opera in due modi: in input e in edit.

INPUT	per immettere dati.
EDIT	per immettere i sottocomandi e i loro operandi.

All'inizio viene specificato il nome del data-set con il comando **EDIT**.

Se viene specificato l'operando **NEW** automaticamente si entra in **INPUT**.

Se non viene specificato **NEW** allora si entra in **EDIT** se il data-set esiste altrimenti in **INPUT**; anche se il data-set esiste ma è vuoto si viene messi in **INPUT**.

In **EDIT** si può lavorare riferendosi al numero di linea (**line-number editing**) oppure riferendosi a una parte specifica del testo (**context editing**).

SOTTOCOMANDI EDIT

BOTTOM

B

CHANGE

C

[line-number-1 [line-number-2]
string-1 [string-2 [ALL]]]

DELETE

DEL

[line-number-1 [line-number-2]]

DOWN

D

[count]

END

[SAVE]

[NOSAVE]

FIND [string]
F

HELP

INPUT [line-number-1 [increment]]
I [R]
[I]

INSERT [inser data]
IN

insert/replace/delete

LIST [line-number-1 [line-number-2]]
L [NUM]
[SNUM]

RENUM [new-line-number [increment [old-line-number
REN [end-line-number]]]]

RUN
R

SAVE [dsname]
S

TABSET [ON [(integer-list)]]
TAB [OFF]
[IMAGE]

TOP

UP [count]

VERIFY [ON]
V [OFF]

BOTTOM

serve per posizionarsi sull'ultima linea del data-set che si sta editando.

CHANGE

occorre quando si deve modificare una sequenza di caratteri in una linea o in più linee.

Le stringhe di caratteri possono essere delimitate da apostrofi (') o da caratteri non contenuti nella stringa.

Si richiede:

- la prima linea su cui fare il cambiamento;
- il blocco di caratteri da cambiare;
- il blocco di caratteri da rimpiazzare.

Uno dei due blocchi può avere anche valore nullo.

Sono opzionali:

- l'ultima linea che delimita i records su cui

	cercare il blocco di caratteri;
	- ALL che fa cambiare tutti i blocchi di caratteri uguali.
DELETE	si usa per rimuovere uno o più records dal data-set che si sta editando. Seguito dal numero della linea da eliminare. Seguito dal numero della prima linea da eliminare e dal numero dell'ultima linea da eliminare.
DOWN	per spostare il puntatore di linea in quella successiva oppure, seguito da un contatore di linee, viene spostato di n linee quante indicate.
END	serve per terminare la sessione di EDIT . Si può usare questo sottocomando con o senza gli operandi SAVE o NOSAVE per salvare o no il contenuto del data-set modificato.
FIND	per cercare una particolare sequenza di caratteri.
HELP	per ottenere la sintassi e la funzione dei sottocomandi EDIT .
INPUT	per mettere il sistema in INPUT MODE ; il puntatore è posizionato sull'ultima riga immessa. Senza operandi i nuovi dati vengono accodati, con line-number i dati vengono immessi dopo la linea specificata, l'operando R permette il rimpiazzamento dei dati preesistenti, l'operando I permette di inserire linee nuove senza alterare le linee preesistenti.
INSERT	serve per inserire una o più nuove linee di dati nel data-set. Se non si fa seguire il sottocomando dai dati il sistema si mette in INPUT MODE . Possiamo inserire un record alla volta.
insert/replace/delete	questi sottocomandi possono essere utilizzati senza richiamarli direttamente, occorre soltanto, infatti, specificare il numero di linea per ottenere l'effetto voluto: <ul style="list-style-type: none"> - inserendo un nuovo numero di linea seguito da una sequenza di dati automaticamente viene inserita una nuova linea; - dando un numero di linea preesistente seguito da una sequenza di caratteri questi rimpiazzano quelli che c'erano prima; - specificando soltanto il numero di linea questa viene eliminata dal data-set.
LIST	per listare una o più linee del data-set al terminale. Se non si specificano gli operandi viene fatta la lista di tutto il data-set. Specificando il record d'inizio e quello di fine viene listato l'intero blocco di record. Specificando un solo numero viene listato il record avente quel numero di linea. <ul style="list-style-type: none"> - NUM per listare anche il numero di linea (default). - SNUM per listare solo il contenuto del record senza la numerazione.
RENUM	assegna un numero di linea a ciascun record in un data-set senza numerazione. Rinumerazione di ciascun record in un data-set che ha i numeri di linea. <ul style="list-style-type: none"> - Il nuovo numero di linea assegnato è il primo che

	viene messo dopo il sottocomando, se omissso assume 10.
	<ul style="list-style-type: none"> - Incremento dal primo in poi è il secondo che viene messo dopo il sottocomando, se omissso assume 10. - Old-line-number la locazione da dove iniziare la numerazione. - End-line-number il numero di linea dove terminare la numerazione.
RUN	per compilare, caricare e eseguire le istruzioni del data-set che si sta editando.
SAVE	per salvare il data-set in un data-set permanente. Se non viene specificato alcun operando il data-set rimpiazza la versione precedente. Seguito da un nome questo viene assegnato alla versione in corso.
TABSET	serve per stabilire o cambiare le tabulazioni, oppure per cancellare qualsiasi tabulazione esistente. <ul style="list-style-type: none"> - ON senza una lista specificata vuol dire che si accettano i valori di default. (default) - OFF annulla qualsiasi tabulazione. - IMAGE definisce con la prossima linea di input la nuova tabulazione.
TOP	serve per posizionarsi all'inizio del data-set che si sta editando. E' l'inverso del sottocomando BOTTOM .
UP	il puntatore viene spostato indietro di n posizioni, se n è omissso assume 1. E' l'inverso del sottocomando DOWN .
VERIFY	per conoscere la linea corrente, ovvero dove il puntatore è posizionato. Questo comando è attivato da ognuno dei seguenti sottocomandi: BOTTOM , CHANGE , DELETE , DOWN , FIND , RENUM e UP .

FREE

FREE	<p>ALL</p> <p>DSNAME (dataset-name-list)</p> <p>DATASET (dataset-name-list)</p> <p>DDNAME (file-name-list)</p> <p>FILE (file-name-list)</p> <p>ATTRLIST (attr-list-name)</p> <p>[DEST (station-id)]</p> <p>[HOLD]</p> <p>[NOHOLD]</p> <p>[SYSOUT (class)]</p>
-------------	---

Il comando **FREE** viene usato per rilasciare (disallocare) data-sets precedentemente allocati di cui non si ha più bisogno. Può essere usato per rilasciare data-sets allocati dalla procedura di **LOGON**, che non vengono rilasciati automaticamente, data-sets di **SYSOUT** a cui si può cambiare la classe per la stampa oppure fare il **ROUTE** per spedirla ad un altro utilizzatore.

ALL	si richiede la disallocazione di tutti i data-sets allocati dinamicamente.
DSNAME o DATASET	specificano il nome, o i nomi, del data-set che si vuol rilasciare.
DDNAME o FILE	specificano il nome, o i nomi, del file che identifica il data-set da rilasciare.
ATTRLIST	specifica il nome di una o più liste di attributi che si vogliono cancellare.
DEST	specifica il nome di una stazione remota a cui si vuol far arrivare un data-set di SYSOUT .
HOLD	specifica che il data-set è in coda di HOLD (naturalmente relativo al SYSOUT).
NOHOLD	è l'inverso del precedente.
SYSOUT	specifica una classe di OUTPUT , rappresentata da un solo carattere.

LISTALC

LISTALC	[STATUS]
LISTA	[HISTORY]
	[MEMBERS]
	[SYSNAME]

Il comando **LISTALC** viene utilizzato per ottenere una lista dei data-sets allocati durante la corrente sessione di **TSO**.

STATUS	fornisce il nome del data-set e il DDNAME con cui è stato allocato.
HISTORY	si vuol sapere informazioni circa la storia di ciascun data-set.
MEMBERS	si vuole la lista della libreria dei nomi di ciascun data-set partitioned.
SYSNAME	per conoscere tutti i nomi qualificati dei data-sets aventi nomi generati dal sistema.

LISTBC

LISTBC	[MAIL]
LISTB	[NOMAIL]
	[NOTICES]
	[NONOTICES]

Il comando **LISTBC** viene utilizzato per ottenere la lista del contenuto del data-set **SYS1.BROADCAST**, contenente messaggi di interesse generale (**NOTICE**) che vengono spediti dal sistema a tutti i terminali e i messaggi diretti ad un utilizzatore particolare (**MAIL**).

MAIL	specifica che si vogliono ricevere i messaggi dal
-------------	---

NOMAIL
NOTICES
NONOTICES

data-set che erano stati spediti specificatamente per l'utilizzatore. (default)
l'inverso del precedente.
si vogliono ricevere i messaggi spediti a tutti gli utilizzatori. (default)
l'inverso del precedente.

LISTCAT

LISTCAT
LISTC

[CATALOG (catname [/password])]

[CLUSTER]

[NONVSAM]
[NVSAM]

[USERCATALOG]
[UCAT]

Il comando **LISTCAT** viene usato per ottenere una lista di entries dal catalogo.
CATALOG seguito dal nome del catalogo e l'eventuale password, racchiusi fra parentesi, specifica il nome del catalogo che contiene le entries che devono essere listate.
CLUSTER si vogliono listate solo le entries dei cluster.
NONVSAM si vogliono listate le entries dei data-sets non-Vsam.
USERCATALOG si vogliono listate le entries degli user catalog.

LISTDS

LISTDS
LISTD

(data-set-list)

[STATUS]
[HISTORY]
[MEMBERS]
[LABEL]
[CATALOG (cat.name)]
[LEVEL]

Il comando **LISTDS** viene utilizzato per ottenere gli attributi di uno specifico data-set. Il comando lavora indifferentemente per data-sets Vsam e data-sets non-Vsam. Se non si è sicuri di quale tipo di data-set si tratta basta dare il comando senza operandi.

Per i data-sets non-Vsam si ottiene:

- **VOLID** del volume su cui risiede il data-set;
- **LRECL**;
- **BLKSIZE**;
- **RECFM**;
- **DSORG** che può essere:**PS**,
PO,
IS,
DA,

VSAM,

** non specificato,

?? per tutte le altre specificazioni;

- informazioni della directory per i data-sets partitioned;
- la data di creazione, l'expiration date e, solo per i non-Vsam, gli attributi di sicurezza;
- DSCB.

data-set-list	il nome di uno o più data-sets di cui si vogliono le informazioni.
STATUS	si vogliono a terminale anche la DDname associata al data-set e informazioni circa la catalogazione e la possibilità di eliminare il data-set.
HISTORY	si vuole ottenere la data di creazione e di distruzione e se il data-set è protetto o no.
MEMBERS	si vuole la lista di tutti i membri di un data-set partitioned.
LABEL	si vuole listata a terminale la DSCB , la notazione è in esadecimale.
CATALOG	specifica lo user catalog che contiene il nome del data-set.
LEVEL	specifica che i nomi nella lista del data-set sono al più alto livello di qualificatori.

RENAME

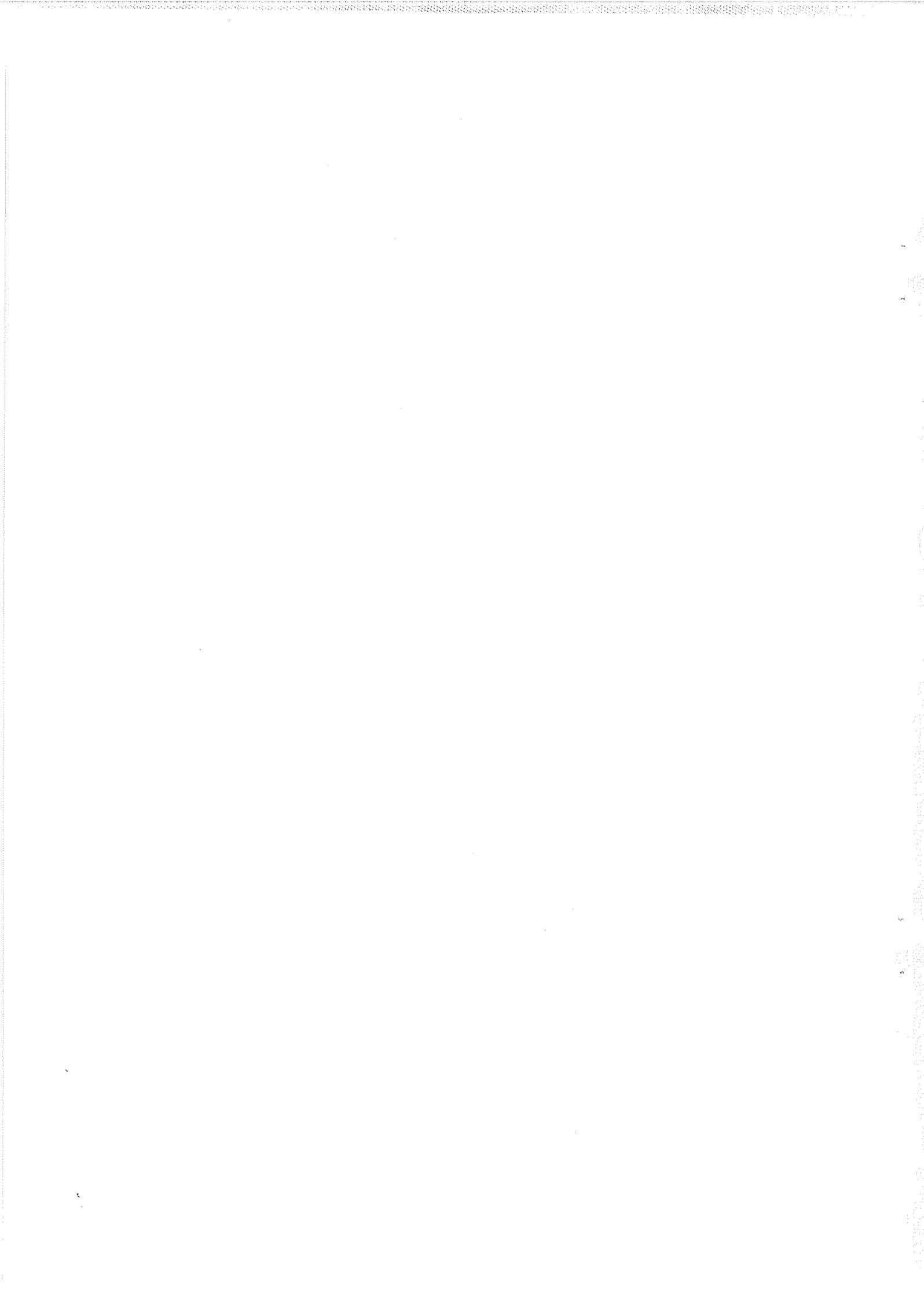
RENAME	old-name	new-name
REN	[ALIAS]	

Il comando **RENAME** viene usato per:

- cambiare il nome a un data-set catalogato non-vsam;
- cambiare il nome a un membro di un data-set partitioned;
- creare un **ALIAS** per un membro di un data-set partitioned.

old-name	è il nome che si vuol cambiare.
new-name	è il nuovo nome da assegnare.
ALIAS	specifica che il nuovo nome viene preso come un alias per il membro identificato con il vecchio nome.

**COMANDI
PER
L'ESECUZIONE
E IL
DEBUG
DEI
PROGRAMMI**



CALL

CALL dsname
dsname (membername)

Il comando **CALL** viene utilizzato per caricare ed eseguire un programma di cui esiste la forma eseguibile (load module creato dal linkage editor). Occorre solo specificare il nome del programma che deve essere eseguito.

dsname specifica il nome di un data-set sequenziale che deve essere eseguito.
dsname (membername) specifica il nome di un membro di un data-set partitioned che deve essere eseguito.

LINK

LINK (data-set-list)

[AMODE (24)
(31)
(ANY)]
[RMODE (24)
(ANY)]

[TERM]
[NOTERM]

L'uso del comando **LINK** invoca il linkage-editor che deve convertire uno o più moduli oggetto (l'output del compilatore) in un modulo eseguibile.

data-set-list specifica il nome di uno o più data-sets contenenti i moduli oggetto.

AMODE specifica il modo di indirizzamento per il modulo che deve essere link-edited.

I valori di **AMODE** sono:

24 per indicare che il modulo deve essere chiamato con indirizzamento a **24-bit**;

31 per indicare che il modulo deve essere chiamato con indirizzamento a **31-bit**;

ANY per indicare che il modulo deve essere chiamato con lo stesso indirizzamento del chiamante.

RMODE specifica il modo di risiedere del modulo che deve essere link-edited.

I valori di **RMODE** sono:

24 per indicare che il modulo deve risiedere sotto la linea dei **16M**;

ANY per indicare che il modulo può risiedere in qualsiasi posizione della memoria.

TERM specifica che si vogliono i messaggi di errore direttamente a terminale. (default)

NOTERM non si vogliono messaggi di errore a terminale ma solo nel data-set di **PRINT**.

LOADGO

LOADGO
LOAD

(data-set-list)
['parameters']
[AMODE(24)
(31)
(ANY)]
[RMODE(24)
(ANY)]
[TERM]
[NOTERM]
[NAME (program-name)]

Il comando **LOADGO** permette di caricare un programma, compilato o assemblato, in memoria reale ed iniziare l'esecuzione. Il comando carica i moduli oggetto prodotti da un compilatore o da un assemblatore e carica i moduli del linkage-editor.

- data-set-list** specifica il nome di uno o più moduli oggetto e/o load moduli per caricarli ed eseguirli. I nomi possono essere nomi di data-sets, nomi di membri di data-sets partitioned o entrambi. Quando si specifica più di un nome, essi devono essere inclusi fra parentesi.
- 'parameters'** specifica qualsiasi parametro da passare al programma per essere eseguito.
- AMODE** specifica il modo di indirizzamento per il modulo che deve essere caricato.
I valori validi per **AMODE** sono:
 24 per indicare che il modulo deve essere chiamato con indirizzamento a **24-bit**;
 31 per indicare che il modulo deve essere chiamato con indirizzamento a **31-bit**;
 ANY per indicare che il modulo deve essere chiamato con lo stesso indirizzamento del chiamante.
- RMODE** specifica il modo di risiedere del modulo che deve essere caricato. Il default è **24**.
I valori validi per **RMODE** sono:
 24 per indicare che il modulo deve risiedere sotto la linea dei **16M**;
 ANY per indicare che il modulo deve risiedere in qualsiasi posizione di memoria.
- TERM** specifica che si vogliono tutti gli errori direttamente a terminale. (default)
- NOTERM** non si vogliono messaggi a terminale ma solo nel data-set di **PRINT**.
- NAME (program-name)** specifica il nome che si vuole assegnare a programma caricato.

RUN

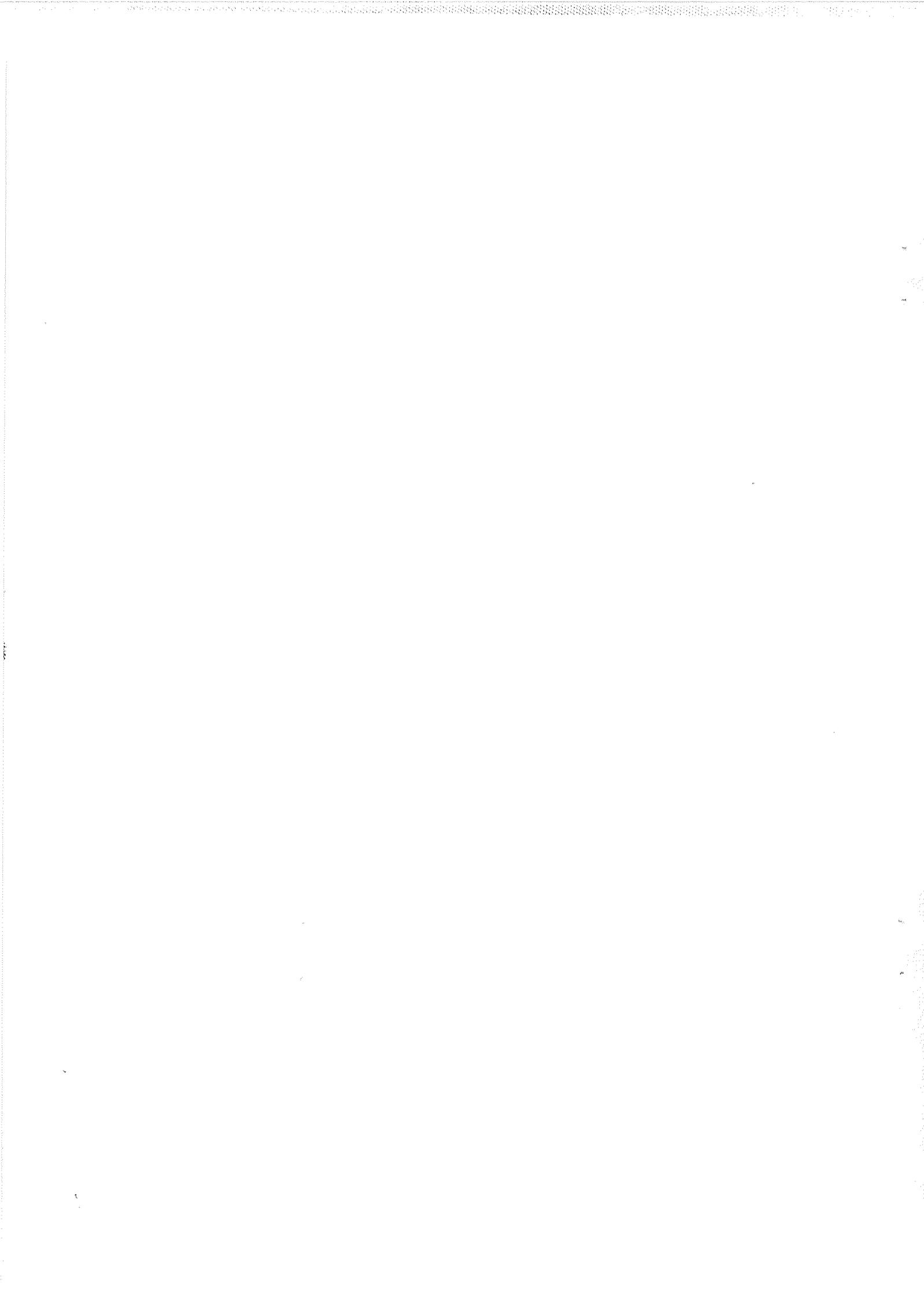
RUN

data-set-name

[ASM [LIB(data-set-list)]]
[COBOL [LIB(data-set-list)]]
[FORT [LIB(data-set-list)]]
[PLI [CHEK] [LIB(data-set-list)]]
[OPT]]
[VSBASIC]

Si usa il comando **RUN** per compilare, caricare ed eseguire un programma sorgente in un data-set. Viene utilizzato per certi programmi prodotto.

- data-set-name** specifica il nome del data-set che contiene il programma sorgente.
- ASM** specifica il programma prodotto del **TSO ASSEMBLER Prompter** e il compilatore dell'**Assembler(F)** che deve essere richiamato per eseguire il programma.
- LIB(data-set-list)** indica la libreria o le librerie che contengono le subroutines necessarie al programma che si vuole far girare.
- COBOL** specifica che il **TSO COBOL Prompter** e il **VSE Cobol** programma prodotto devono essere richiesti per eseguire il programma.
- FORT** specifica il **TSO FORTRAN Prompter** e il **Fortran I (G1)** programma prodotto da richiamare per eseguire il programma.
- PLI** specifica il **PL/I Prompter** e i compilatori **Optimize** o **Checkout** da richiamare per eseguire il programma.
CHECK per il compilatore **Checkout**.
OPT per il compilatore **Optimizing**. (default)
- VSBASIC** specifica che deve essere richiamato il programma prodotto **VSBASIC** per eseguire il programma.



**COMANDI
PER LA
GESTIONE
DEI
PROGRAMMI
BATCH**

CANCEL

CANCEL (jobname [(jobid)]-list)

[NOPURGE]

[PURGE]

Il comando **CANCEL** viene utilizzato per fermare l'esecuzione di un job che è stato sottomesso da terminale. Se il job è stato cancellato con successo a terminale apparirà il messaggio **READY**. Un messaggio comparirà anche sulla console dell'operatore.

jobname	specifica il nome del job che si vuole cancellare. Questo nome deve essere formato dal nome della macchina TSO seguito da uno o più caratteri alfanumerici fino a un massimo di 8 caratteri.
jobid	è l'identificatore assegnato dal JES al momento della sottomissione. Questo può essere utile se vengono sottomessi più job con lo stesso nome.
NOPURGE	viene cancellato solo il job mentre l'output è ancora disponibile. (default)
PURGE	viene cancellato il job e l'output.

OUTPUT

OUTPUT (jobname [(jobid)]-list)

OUT

[CLASS (classname-list)]

[PRINT (dsname)]

[KEEP]

[NOKEEP]

[DELETE]

[DEST]

[NEWCLASS]

Il comando **OUTPUT** invia l'output da un job al terminale, da un job ad uno specifico data-set, cancella l'output dei jobs, cambia la classe di output per i jobs e, infine, fa il route dell'output di un job a una stazione remota.

jobname	specifica il nome di uno o più jobs di cui vogliamo trattare l'output.
jobid	specifica l'identificatore o gli identificatori dei jobs di cui vogliamo trattare l'output.
CLASS	specifica il nome della classe da ricercare per l'output.
PRINT	specifica il nome del data-set in cui deve andare a finire l'output. Se è seguito da un asterisco l'output finisce a terminale.
KEEP	significa che l'output deve rimanere nelle code in attesa di stampa.
NOKEEP	è l'inverso del precedente. (default)

DELETE specifica che le classi di output specificate con l'operando **CLASS** devono essere distrutte.
DEST invia le classi di **SYSOUT** a una stazione remota.
NEWCLASS per cambiare le classi a uno o più **SYSOUT**.

SUBMIT

SUBMIT (data-set-list)
SUB

Il comando **SUBMIT** viene usato per sottomettere uno o più jobs per l'esecuzione batch.

data-set-list specifica uno o più nomi di data-sets o nomi di membri di data-sets partitioned che devono essere sottomessi per l'esecuzione.

STATUS

STATUS (jobname [(jobid)]-list)
ST

Il comando **STATUS** fornisce lo stato di uno o più jobs batch a terminale. E' possibile ottenere lo stato di tutti i jobs, di un certo numero di determinati jobs o di un singolo job. La risposta può essere: in attesa di esecuzione, in esecuzione o in attesa di output.

jobname o **jobid** specifica il nome o l'identificazione del, o dei, job di cui si vuol conoscere lo stato.