TABLE OF CONTENTS

1. WHAT IS CYTOS? 1-00-01

A. CAPABILITIES OF CYTOS/I 1-00-01

2. HOW TO USE THE CYTOS SYSTEM -- BASIC CONCEPTS

A. INVOKING THE CYTOS/I SYSTEM 2-00-01
B. TRANSMITTING INFORMATION TO THE SYSTEM 2-00-01
C. COMMAND LANGUAGE 2-00-04
D. BASIC COMMANDS AND SYSTEM CONVENTIONS 2-00-04

(1) THE EXPLAIN COMMAND 2-00-04
(2) STORING DATA AND PROGRAMS 2-00-05
(3) THE SYSTEM EDITOR 2-00-07
(4) ANSWERING THE WHATFILE QUESTION 2-00-08
(5) ENDING A CYTOS SESSION -- LOGOFF 2-00-09
(6) SAMPLE SESSIONS 2-00-10

3. INTERACTIVE PROGRAMMING TECHNIQUES 3-00-01

4. USER PROGRAMS

A. CREATING A SOURCE PROGRAM 4-00-01
B. COMPILING THE SOURCE PROGRAM 4-00-02
C. CREATING A LOAD-MODULE 4-00-03
D. EXECUTING A USER-WRITTEN COMMAND 4-00-04
E. THE DEBUGGING PROCESS 4-00-04

5. EDITOR TUTORIAL

A. USES OF THE EDITOR 5-00-01
B. COMMAND FORMAT 5-00-01
C. BASIC CONCEPTS 5-00-01
D. EDITOR COMMAND PARAMETERS 5-00-03
E. MULTIPLE COMMAND FACILITY 5-00-09
F. EDITOR COMMANDS 5-00-10

AFTER 5-00-11
ANCHOR 5-00-11
ATTN 5-00-12
BEFORE 5-00-12
BOTTOM 5-00-12
CAPITALIZE 5-00-13
CHANGE 5-00-13
CWALLS 5-00-13
DEFINE 5-00-15
DELETE 5-00-16
DISPLAY 5-00-17
DO 5-00-17

TABLE OF CONTENTS

END	5-00-18
EXPLAIN	5-00-18
FILE	5-00-18
FOLD	5-00-19
INFL	5-00-19
INSERT	5-00-20
LABEL	5-00-21
LL	5-00-21
LOCATE	5-00-21
NEXT	5-00-22
OVERLAY	5-00-22
PL	5-00-22
PRINT	5-00-23
PWALLS	5-00-23
QUIT	5-00-23
REPLACE	5-00-24
SAVE	5-00-24
SET	5-00-24
TABS	5-00-26
TOP	5-00-26
UPDATE	5-00-27
USE	5-00-27
UTILITY	5-00-27
WHATS	5-00-28
'PROCEDURE_NAME'	5-00-28

6. PERMISSION TUTORIAL 6-00-01

7. UTILITY TUTORIAL 7-00-01

8. CARD-INPUT MODE

A. THE CONCEPT OF A CYTOS JOB	8-00-01
B. SETTING UP A CARD-INPUT JOB	8-00-02
C. SPECIAL CONSIDERATIONS	8-00-03
D. SPECIAL CARD-INPUT PSEUDO-COMMANDS	8-00-04
.ATTENTION	8-00-04
.END	8-00-05
.RESET	8-00-05
E. A WORD ABOUT 'EXECUTE-MODE'	8-00-06

9. CYTOS CONVENTIONS AND COMMANDS

A. DATA SET NAMING CONVENTIONS	9-01-01
B. FORMAT OF COMMAND DESCRIPTIONS	9-02-01
C. CYTOS/I SYSTEM COMMAND DESCRIPTIONS	9-03-01

TABLE OF CONTENTS

COMMAND	DM2 EQUIVALENT	
ALLOCATE	(*DFILE)	9-03-01
ASSEMBLE	(// ASM)	9-03-02
ASSIGN		9-03-04
BATCH		9-03-05
CATALOGUE	(*STORE WS UA)	9-03-05
CATPRINT	(*DUMPLET/FLET)	9-03-06
CHECKPOINT		9-03-07
CLEAR		9-03-09
COBOL	(// COBOL)	9-03-10
COMMA		9-03-11
CONTINUE		9-03-12
COPY	(*DUMP,*STORE)	9-03-13
DCOM		9-03-13
DESCRIBE	(*DUMPLET NAME)	9-03-14
DUMP	(*DUMP UA PR)	9-03-14
EDIT		9-03-15
EJECT	(// EJECT)	9-03-17
ERASE	(*DELETE)	9-03-17
EXECUTE		9-03-18
EXPLAIN		9-03-20
FIXLIB	(*DEFINE FIXED AREA)	9-03-21
FLUSH		9-03-21
FORTRAN	(// FOR)	9-03-22
GRAPHICS	(*G2250)	9-03-25
INCLUDE	(*NOCAL)	9-03-26
JOIN		9-03-28
KEYBOARD		9-03-29
LINKEDIT	(*STORECI)	9-03-30
LIST	(*STOREDATAE,*DUMPDATAE)	9-03-32
LOADGO	(// XEQ DSF)	9-03-33
LOGOFF		9-03-34
LOGON		9-03-35
MACLIB	(*MACRO UPDATE)	9-03-36
MOUNT	(// JOB)	9-03-36
MOVE	(*DUMP,*STOREMOD)	9-03-38
NAMEFILE	(*FILES)	9-03-39
NEWPASS		9-03-40
NEWSPACE	(*STOREDATA WS UA)	9-03-40
OVERLAY	(*LOCAL)	9-03-42
PERMIT		9-03-43
PRINT	(*DUMPDATAE UA PR)	9-03-44
PUNCH	(*DUMPDATAE UA CD)	9-03-45
READ	(*STOREDATAE CD UA)	9-03-46
RECATALOGUE	(*STOREMOD WS UA)	9-03-47
REFORMAT	(*DWADR)	9-03-48
REMOVE	(*DUMP UA CD)	9-03-48
RENAME		9-03-49
REPRODUCE	(*STOREDATAE,*DUMPDATAE CD)	9-03-50
RESTART		9-03-50

TABLE OF CONTENTS

RESTORE	(*STORE(DATA(CI)) CD UA)	9-03-51
RESUME		9-03-52
RPG	(// RPG)	9-03-53
RUN	(// XEQ DCI)	9-03-54
SECTOR		9-03-56
SET		9-03-57
SUBSTITUTE	(*EQUAT)	9-03-65
TYPE		9-03-59
UNCATALOGUE	(*DUMP UA WS,*DELETE)	9-03-60
UNPERMIT		9-03-60
VOID	(*DEFINE VOID)	9-03-61
WORKON	(*DUMP UA WS)	9-03-61
XF	(*FILES - XBASE)	9-03-62
ZAP	(// XEQ MODIF)	9-03-63

10. TIPS AND TECHNIQUES

A. ON USING THE 'FIXED AREA' FACILITY	10-00-01
B. ON INTERRUPTING AND RESUMING COMMANDS	10-00-02
C. ON DYNAMIC VOLUME MOUNTING	10-00-04
D. ON OPTIMIZING DISK VOLUME SPACE	10-00-04
E. ON OPTIMIZING DISK ACCESS TIME	10-00-05
F. ON REPRODUCING BINARY DECKS	10-00-06
G. ON MAINTAINING MACRO LIBRARIES	10-00-06
H. ON RUNNING IBM UTILITIES UNDER CYTOS	10-00-07
I. ON EDITING LARGE FILES	10-00-07
J. ON EDITING LARGE CARD FILES	10-00-08
K. ON DEFINING COMPLEX EDITOR PROCEDURES	10-00-08
L. ON SAVING EDITOR DEFINITIONS	10-00-09

11. APPENDICES

A. CYTOS COMMAND ATTRIBUTES	11-00-01
B. DATA SET ATTRIBUTES, BY COMMAND	11-00-04
C. CYTOS/I WAIT CODES	11-00-07

1. FACILITIES OF THE CYTOS SYSTEM

A. CAPABILITIES OF CYTOS/I

CYTOS/I IS A SERIES OF EXTENSIONS AND MODIFICATIONS TO THE 1130 DISK MONITOR SYSTEM (VERSION 2) TO SUPPORT ONE INTERACTIVE TERMINAL: THE SYSTEM KEYBOARD/CONSOLE PRINTER. THE MAJOR DESIGN REQUIREMENT HAS BEEN THAT ANY PROGRAM THAT RUNS UNDER DMS SHOULD ALSO RUN UNDER CYTOS, GIVEN ENOUGH CORE. THUS THE FORTRAN AND RPG COMPILERS ARE THE SAME FOR BOTH BATCH AND TERMINAL USE.

CYTOS IS A CONVERSATIONAL SYSTEM. THIS MEANS THAT AS WELL AS BEING ABLE TO SET UP A COMPLETE PROGRAM DECK AND DATA BEFORE SUBMITTING A JOB TO BE RUN, THE TERMINAL USER CAN INTERACT WITH THE COMPUTER WHILE HIS JOB IS PROCESSED. HE CAN ENTER DATA WHEN HIS PROGRAM REQUIRES IT, SEE OUTPUT AS IT IS PRODUCED, STOP EXECUTION OF A PROGRAM THAT IS NOT PRODUCING THE DESIRED RESULTS, START IT AGAIN WITH DIFFERENT INPUT PARAMETERS, OR CORRECT HIS PROGRAM'S ERRORS AND RECOMPILE IT. CYTOS CAN ALSO BE RUN IN A BATCH ENVIRONMENT BY SUBMITTING CYTOS COMMANDS ON CARDS (SEE SECTION 8).

CYTOS, LIKE MANY OTHER CONVERSATIONAL SYSTEMS, ALLOWS THE USER TO STORE DATA OR PROGRAMS IN THE SYSTEM. INFORMATION STORED IN THE SYSTEM IS SAID TO BELONG TO A 'DATA SET'. A DATA SET IS ANY NAMED COLLECTION OF INFORMATION THAT THE USER WANTS TO CONSIDER A LOGICAL UNIT. A DATA SET MAY CONTAIN A SOURCE PROGRAM, AN OBJECT PROGRAM, DATA THAT IS TO BE USED BY A PROGRAM, OR DATA THAT IS PRODUCED BY A PROGRAM. A USER MAY SHARE HIS DATA SETS WITH OTHER USERS.

BESIDES BEING ABLE TO ENTER DATA AT THE TERMINAL AND RECEIVE RESULTS THERE, THE USER MAY SPECIFY AT THE TERMINAL THAT HE WANTS TO SCHEDULE A DATA SET TO BE PRINTED ON THE SYSTEM PRINTER OR PUNCHED ON THE SYSTEM CARD PUNCH. HE MAY ALSO SUBMIT A CARD DECK TO BE READ INTO A DATA SET FROM THE SYSTEM CARD READER. MANY SUCH UTILITY FUNCTIONS ARE AVAILABLE.

CYTOS/I IS NOT IN ITSELF A TIME-SHARING SYSTEM, ALTHOUGH BY DEFINITION IT WILL SUPPORT ANY USER-WRITTEN PACKAGE WHICH CONTROLS MULTIPLE TERMINALS IN A TIME-SHARING MODE, ASSUMING THAT PACKAGE OPERATES UNDER DMS IN A STANDARD MANNER.

2. HOW TO USE THE CYTOS SYSTEM

A. INVOKING THE CYTOS/I SYSTEM

ALTHOUGH CYTOS ACTS AS IF IT WERE IN CONTROL OF THE ENTIRE DISK MONITOR SYSTEM, IT REALLY IS DESIGNED AS A SUB-SYSTEM OF THE MONITOR. AS SUCH, IT MAY BE INVOKED IN THE SAME MANNER AS ANY OTHER PROCESSOR, NAMELY BY PLACING THE APPROPRIATE CONTROL CARD IN THE JOB STREAM. THE CARD THAT INVOKES CYTOS,

```
// CYTOS
```

NEED NOT FOLLOW A '// JOB' CONTROL RECORD, OR ANY OTHER CONTROL RECORD; IT MAY IN FACT FOLLOW DIRECTLY BEHIND THE 'COLD-START CARD' IF NO BATCH PROCESSING IS DESIRED BEFORE CYTOS IS INVOKED.

IF YOU WISH TO RETURN CONTROL TO THE DISK MONITOR SYSTEM AT ANY TIME, YOU NEED ONLY ISSUE THE CYTOS BATCH COMMAND (EXPLAINED IN GREATER DETAIL IN A LATER SECTION OF THIS MANUAL). NOTE: WHEN CYTOS IS RESIDENT ON ANY SYSTEM CARTRIDGE, YOU WILL NOT BE ABLE TO USE THE '// CPRNT' MONITOR CONTROL RECORD, AS THE SLOT FOR THIS RECORD HAS BEEN REQUISITIONED FOR THE '// CYTOS' RECORD. THE '// CPRNT' RECORD, IN ANY CASE, PROVIDES ONLY THE FACILITY THAT CYTOS INCORPORATES AS ITS 'MODUS OPERANDI': COMMUNICATION BETWEEN YOU AND THE SYSTEM VIA THE CONSOLE TYPEWRITER.

THE '// CYTOS' CONTROL RECORD WILL ACTIVATE THE CYTOS COMMAND LANGUAGE INTERPRETER. AFTER THE CARD HAS BEEN READ, YOU WILL BE ASKED TO LOGON. IF THE CYTOS SYSTEM HAS NEVER BEEN USED ON THE CURRENTLY DEFINED SYSTEM CARTRIDGE, IT MUST FIRST BE INITIALIZED USING THE JOIN COMMAND; OTHERWISE, YOU MAY RESPOND BY ISSUING THE LOGON COMMAND ITSELF.

IF THE 'USERID' AND 'PROJID' PARAMETERS TO THE LOGON ARE VALID, THE SYSTEM WILL PROMPT FOR A PASSWORD. WHEN THE PASSWORD HAS BEEN VERIFIED, LOGON WILL BE COMPLETE, AND YOU MAY CONTINUE TO ENTER CYTOS COMMANDS IN ANY ORDER DESIRED.

THE PASSWORD MAY BE CHANGED AT A LATER POINT BY ISSUING THE NEWPASS COMMAND.

B. TRANSMITTING INFORMATION TO THE SYSTEM

WHEN CYTOS IS READY TO ACCEPT A COMMAND, IT WILL PROMPT BY TYPING THE CHARACTERS '==>' AND THEN TABBING OVER SEVERAL CHARACTERS IN ORDER THAT YOU MAY SEE THE PROMPT. WHEN THE WHITE 'KB SELECT' LIGHT TO THE LEFT OF THE KEYBOARD IS LIT, YOU MAY BEGIN TO TYPE A COMMAND. (A CONVENIENT SETTING FOR THE TAB AFTER THE ARROW IS APPROXIMATELY TEN COLUMNS TO THE RIGHT OF THE LEFT MARGIN).

UP TO 80 CHARACTERS MAY BE ENTERED PER LINE OF INPUT. THE 'EOF' KEY MUST BE DEPRESSED BEFORE ANY OF THE INFORMATION TYPED IN THE CURRENT LINE WILL BE ACCEPTED BY CYTOS. IF YOUR LINE REQUIRES TYPING MORE THAN 80 CHARACTERS, YOU MAY CONTINUE INPUT IN SUCCESSIVE RECORDS BY TYPING AN '&' AFTER THE LAST WORD BEFORE PRESSING THE 'EOF' KEY. CYTOS WILL INDICATE THAT IT IS READY TO ACCEPT FURTHER INPUT TO THE CURRENT LINE BY PROMPTING WITH THE CHARACTERS '&=>' INSTEAD OF '==>'.

IF NO INFORMATION IS ENTERED BEFORE PRESSING THE 'EOF' KEY, CYTOS WILL PROMPT AGAIN. THIS THE 'KB SELECT' LIGHT ALWAYS INDICATES WHEN YOU MAY ENTER A COMMAND.

TYPING ERRORS MAY BE CORRECTED BEFORE TRANSMISSION TO CYTOS IN ONE OF THE FOLLOWING WAYS:

1. TO DELETE AN ENTIRE LINE, YOU MAY EITHER PRESS THE 'ERASE FIELD' KEY (STANDARD TO THE 1130 KEYBOARD), OR TYPE THE CHARACTER '¢'. AFTER EITHER OF THESE ACTIONS, YOU MAY BEGIN RE-ENTERING THE LINE IMMEDIATELY, OR YOU MAY PRESS THE 'EOF' KEY AND ENTER THE CORRECTION ON THE NEXT LINE. ONLY THE CHARACTERS AFTER THE '¢' ARE TRANSMITTED.

FOR EXAMPLE:

==> FORTRUN¢FORTRAN MYPROG

IF THE 'ERASE FIELD' KEY IS TYPED, INFORMATION IS ERASED ONLY FOR THE CURRENT RECORD; IF YOU ARE CURRENTLY ENTERING A CONTINUATION RECORD, ONLY THE INFORMATION IN THE CONTINUATION RECORD IS ERASED -- ANY INFORMATION PREVIOUSLY TRANSMITTED TO CYTOS BY PUSHING THE 'EOF' KEY WILL NOT BE AFFECTED. IF THE '¢' IS USED, ON THE OTHER HAND, ALL INFORMATION IN THE CURRENT PROMPT WILL BE ERASED, AND YOU WILL BE STARTING AS IF CYTOS HAD JUST TYPED THE '==>' PROMPT.

2. TO DELETE THE IMMEDIATELY PRECEDING CHARACTER(S), TYPE '@' FOR EACH CHARACTER TO BE DELETED, OR PRESS THE BACKSPACE KEY (<-- ON THE 1130 KEYBOARD) ONCE FOR EACH CHARACTER TO BE DELETED.

FOR EXAMPLE:

==> FORTRUN@@AN MYPROG

THE WORDS FORTRAN MYPROG WILL BE TRANSMITTED. AGAIN, THE DIFFERENCE BETWEEN THE TWO WAYS OF DELETING A CHARACTER IS THAT YOU MAY DELETE ALL THE WAY BACK TO THE BEGINNING OF THE PROMPT BY USING THE '@' CHARACTER; IF YOU USE THE BACKSPACE KEY, ONLY CHARACTERS FOLLOWING THE LATEST 'EOF' KEY MAY BE DELETED.

3. IF A TYPING ERROR IS NOT DETECTED BEFORE THE LINE IS TRANSMITTED, THE SYSTEM WILL ACT ACCORDING TO WHAT WAS TRANSMITTED:

(A). AT CYTOS 'COMMAND LEVEL' (I.E. WHEN THE COMMAND LANGUAGE INTERPRETER IS EXAMINING THE LINE IN ORDER TO DETERMINE WHAT ACTION HAS BEEN REQUESTED), THE SYSTEM MAY DETECT AN ERROR IN FORMAT OR THE MISSPELLING OF THE COMMAND VERB OR OF SOME KINDS OF PARAMETERS. MISSPELLING OF A DATA SET NAME, HOWEVER, WILL BE DETECTED ONLY IF THE INCORRECT DATA SET NAME CANNOT BE USED BY THE COMMAND. WHEN THE SYSTEM DETECTS AN ERROR, IT WILL TYPE BACK AN ERROR MESSAGE, THEN PROMPT FOR A NEW COMMAND. CYTOS ERROR MESSAGES MAY BE CLARIFIED BY USING THE EXPLAIN COMMAND. ALL OTHER ERROR MESSAGES ARE THOSE GIVEN BY THE DISK MONITOR SYSTEM, AND EXPLANATIONS OF THEM WILL BE FOUND IN 'IBM 1130 DISK MONITOR SYSTEM, VERSION 2, PROGRAMMING AND OPERATOR'S GUIDE'.

(B). IF THE LINE WAS TEXT BEING ENTERED WITH THE EDITOR, APPROPRAITE EDITOR COMMANDS MAY BE USED TO CORRECT THE LINE.

(C). IF THE TRANSMITTED LINE WAS INPUT REQUESTED BY A COMMAND IN EXECUTION, ACTION TAKEN WILL DEPEND ON THE PROGRAM. IF THE PROGRAM DOES NOT PROVIDE ERROR PROCESSING, EXECUTION MAY CONTINUE WITH INVALID RESULTS OR THE PROGRAM MAY TERMINATE ABNORMALLY.

ANY CYTOS COMMAND MAY BE GIVEN ONCE THE USER HAS SUCCESSFULLY LOGGED ON TO THE SYSTEM. THE SYNTAX DESCRIBED ABOVE APPLIES EQUALLY TO COMMANDS TYPED IN FROM A TERMINAL UNDER CYTOS/II. THE FUNCTIONAL EQUIVALENT OF THE 1130 INT REQ KEY IS THE ATTN KEY ON A 2741, AND THE BREAK KEY ON A TELETYPE. IT MAY BE DEPRESSED AT ANY TIME THE USER WISHES TO ABORT A PARTICULAR FUNCTION IN ORDER TO BE IMMEDIATELY PROMPTED FOR ANOTHER CYTOS COMMAND. INPUT MAY BE EDITED AS ENTERED BY USING THE LOGICAL ERASE FIELD AND BACKSPACE CHARACTERS (RESPECTIVELY THE CENT-SIGN AND AT-SIGN) AS UNDER STANDARD CYTOS. PHYSICAL KEYS ON THE TERMINAL MAY NOT BE USED FOR THIS PURPOSE UNLESS THE SET COMMAND HAS BEEN GIVEN TO DEFINE THEM AS THE LOGICAL CHARACTERS. THIS IS BECAUSE SOME TERMINALS LACK ONE OR BOTH KEYS; THE TELETYPE, FOR INSTANCE, HAS NO EQUIVALENT OF THE 1130 ERASE FIELD KEY.

WHEN THE USER HAS COMPLETED PROCESSING, HE MAY DISCONNECT HIS TERMINAL FROM THE COMPUTER BY GIVING THE LOGOFF COMMAND. A LOGOFF COMMAND IS AUTOMATICALLY GENERATED BY THE SYSTEM IF FOR SOME REASON THE TERMINAL BECOMES PHYSICALLY DISCONNECTED BECAUSE OF TELEPHONE LINE DIFFICULTIES. THE CONNECTION IS ALSO BROKEN AUTOMATICALLY IF THE USER FAILS TO LOGON PROPERLY WITHIN FIVE MINUTES OF HAVING DIALED INTO THE COMPUTER, OR IF, ONCE SUCCESSFULLY LOGGED ON TO THE SYSTEM, HE FAILS TO GIVE A COMMAND FOR MORE THAN 45 MINUTES.

C. COMMAND LANGUAGE

A 'COMMAND' IS AN INSTRUCTION TELLING THE CYTOS SYSTEM WHAT TO DO. ENTERING A COMMAND IS LIKE USING MONITOR CONTROL CARDS IN THE DISK MONITOR SYSTEM: A COMMAND MAY INSTRUCT THE SYSTEM TO COMPILE A FORTRAN PROGRAM, TO PRINT A DATA SET, TO CHANGE A PASSWORD, OR ANY ONE OF NUMEROUS OTHER FUNCTIONS.

THE FIRST WORD OF EVERY COMMAND IS A COMMAND NAME (COMMAND 'VERB') SPECIFYING A FUNCTION TO BE PERFORMED. MANY COMMANDS REQUIRE ADDITIONAL INFORMATION ('PARAMETERS') TO PERFORM THE FUNCTION. COMMAND VERBS AND PARAMETERS ARE SEPARATED BY ONE OR MORE BLANKS. FOR EXAMPLE:

```
==> EDIT MYPROG FORTRAN
```

IS A COMMAND CONSISTING OF THE COMMAND VERB EDIT AND THE PARAMETERS MYPROG AND FORTRAN.

THE FORMAT OF EACH COMMAND AND A DESCRIPTION OF WHAT IT DOES MAY BE FOUND IN SECTION 9 OF THIS MANUAL. YOU SHOULD BECOME FAMILIAR WITH THE LOGON COMMAND BEFORE YOU ATTEMPT TO USE THE CYTOS SYSTEM FOR THE FIRST TIME, AS YOU WILL NEED TO USE IT TO ACTIVATE THE SYSTEM FOR FURTHER USE.

THE CYTOS SYSTEM IS 'COMMAND-DRIVEN'. THIS MEANS THAT AFTER YOU ENTER A COMMAND, THE SYSTEM WILL PERFORM THE REQUESTED FUNCTION INSOFAR AS POSSIBLE, AND THEN PROMPT FOR YOU TO ENTER ANOTHER COMMAND. ONLY THE CYTOS COMMAND LANGUAGE INTERPRETER CAN ACT UPON SYSTEM AND USER COMMANDS. A COMMAND CAN BE ENTERED ONLY WHEN THE 'KB SELECT' LIGHT IS ON AND THE KEYBOARD HAS BEEN UNLOCKED, BOTH CONDITIONS OCCURRING AFTER THE '==>' HAS BEEN TYPED BY THE SYSTEM. TO STOP AN EXECUTING COMMAND, YOU NEED ONLY PRESS THE 'INT REQ' KEY ON THE CONSOLE KEYBOARD; THIS WILL CAUSE THE SYSTEM TO INTERRUPT THE EXECUTING PROGRAM AND RETURN TO COMMAND LEVEL. (THE COMMAND IS ONLY SUSPENDED; IT MAY BE CONTINUED BY ISSUING THE RESUME COMMAND). AT CRITICAL TIMES, FOR EXAMPLE WHEN DATA SETS ARE BEING ERASED FROM THE SYSTEM, YOU WILL NOT BE ABLE TO INTERRUPT THE COMMAND; THE KEYBOARD SIMPLY WILL NOT UNLOCK UNTIL THE CRITICAL FUNCTION HAS CONCLUDED.

D. SOME BASIC COMMANDS AND SYSTEM CONVENTIONS

1. THE EXPLAIN COMMAND

USE THE EXPLAIN COMMAND TO OBTAIN A DESCRIPTION OF SYSTEM FACILITIES AND ERROR MESSAGES, SUCH AS:

A. CYTOS COMMANDS
B. SYSTEM ERROR CODES
C. LANGUAGE PROCESSOR DIAGNOSTICS
D. CYTOS REVISIONS AND EXTENSIONS

COMMAND FORM:

EXPLAIN (TERM1 TERM2 ... TERM_N)

EACH TERM MAY BE MODIFIED BY A KEYWORD, IN THE FORM 'TERM.SUBTERM'. THE SUBTERM LIMITS THE EXPLANATION TO A SPECIFIC PORTION OF THE SUBJECT SYMBOLIZED BY 'TERM'.

EXAMPLES:

A. ==> EXPLAIN EXPLAIN
DESCRIBES THE EXPLAIN COMMAND.

B. ==> EXPLAIN CYTOS.COMMANDS
LISTS CURRENTLY IMPLEMENTED CYTOS COMMANDS.

C. ==> EXPLAIN FORTRAN.LIST
DESCRIBES THE LIST OPTION IN THE FORTRAN COMMAND.

D. ==> EXPLAIN (CL03 D21 F007)
PROVIDES INFORMATION RELATIVE TO THE THREE ERROR MESSAGES WHOSE IDENTIFICATIONS ARE LISTED IN THE PARAMETER.

2. STORING DATA AND PROGRAMS

YOU MAY STORE PROGRAMS AND/OR DATA IN NAMED COLLECTIONS OF LOGICALLY RELATED INFORMATION, CALLED 'DATA SETS'. THESE DATA SETS USUALLY RESIDE ON A DIRECT-ACCESS STORAGE DEVICE, BUT YOU MAY REMOVE THEM TO CARDS BY USING THE REMOVE COMMAND.

YOU ARE THE 'AUTHOR' OF A DATA SET IF YOU CREATE IT AND STORE IT IN YOUR LIBRARY. FOR PURPOSES OF DEFINITION, A LIBRARY IS A DIRECT-ACCESS STORAGE VOLUME (I.E. DISK CARTRIDGE) WHICH BELONGS TO YOU BY VIRTUE OF YOUR HAVING ESTABLISHED OWNERSHIP BY ISSUING THE JOIN COMMAND. WHEN ISSUING CYTOS COMMANDS, YOU MAY REFER TO YOUR OWN DATA SETS, TO DATA SETS BELONGING TO OTHER USERS (IF PERMISSION HAS BEEN GRANTED YOU BY THOSE USERS -- SEE THE PERMIT COMMAND), OR TO DATA SETS BELONGING TO THE CYTOS SYSTEM.

(A). DATA SET NAMES

TO FULLY IDENTIFY A DATA SET YOU MUST USE THE 'FULLY QUALIFIED DATA SET NAME', SUCH AS

BROWN.PROGM.S

WHERE BROWN IS THE AUTHOR/OWNER OF THE DATA SET, PROGM IS THE SIMPLE NAME OF THE DATA SET, AND S IS THE TYPE.

SIMPLE DATA SET NAMES (DSNAMES) MUST BE FROM ONE TO EIGHT ALPHANUMERIC CHARACTERS, BEGINNING WITH AN ALPHABETIC CHARACTER. THE SYSTEM USES ONLY THE FIRST FIVE CHARACTERS OF THE NAME, SO THESE MUST BE UNIQUE FOR ALL DATA SETS OF THE SAME TYPE. (EIGHT CHARACTERS ARE ALLOWED IN ORDER TO MAINTAIN EXTERNAL COMPATIBILITY WITH OTHER VERSIONS OF CYTOS). THE CHARACTERS '$' AND '#' ARE VALID WITHIN DATA SET NAMES.

THERE ARE THREE TYPES OF DATA SETS:

.S, FOR SOURCE PROGRAMS OR DATA;
.O, FOR OBJECT PROGRAMS (SEE NOTE 1);
.L, FOR LOAD MODULES (SEE NOTE 2).

NOTE 1: THE OUTPUT OF A COMPILER, CALLED AN 'OBJECT MODULE', IS IN MACHINE LANGUAGE, BUT IS NOT YET EXECUTABLE. IT MAY STILL CONTAIN REFERENCES TO SYMBOLIC NAMES (SUCH AS SUBROUTINES) THAT WERE NOT DEFINED IN THE SOURCE PROGRAM. OBJECT MODULES ARE IN DISK SYSTEM FORMAT.

NOTE 2: THE EXECUTABLE PROGRAM PRODUCED BY THE CORE LOAD BUILDER IS CALLED A 'LOAD MODULE.' THE CORE LOAD BUILDER RESOLVES ALL EXTERNAL REFERENCES BY COMBINING THE GIVEN OBJECT MODULES AND BY SEARCHING SUBROUTINE LIBRARIES. LOAD MODULES ARE IN DISK CORE-IMAGE FORMAT.

WHEN YOU REFER TO YOUR OWN DATA SETS, YOU NEED NOT QUALIFY THE DSNAME WITH YOUR USERID. THE SYSTEM WILL ASSUME THAT YOU ARE REFERRING TO YOUR OWN DATA SETS UNLESS YOU SPECIFY SOMEONE ELSE'S.

IN SOME COMMANDS IT IS CLEAR BY IMPLICATION WHAT TYPE OF DATA SET IS BEING REFERRED TO. IN SUCH CASES YOU NEED NOT SPECIFY THE TYPE OF DATA SET. HOWEVER, IF YOU WISH TO REFER TO SOMEONE ELSE'S DATA SET, YOU MUST QUALIFY THE SIMPLE DSNAME WITH BOTH THE USERID AND THE DATA SET TYPE:

SMITH.CNVRT.O

SEVERAL CYTOS COMMANDS ALLOW YOU TO ENTER DATA DIRECTLY FROM THE TERMINAL OR FROM CARDS RATHER THAN FROM A DISK DATA SET (E.G. FORTRAN). IN SUCH INSTANCES YOU MAY REFER TO THE TERMINAL BY THE SYMBOLIC NAME '.T' (OR '.TERMINAL'), AND TO THE PRINCIPAL INPUT/OUTPUT UNIT (WHETHER CARD READ/PUNCH OR PAPER TAPE READ/PUNCH) WITH THE NAME '.C' (OR '.CARDS'). IN SOME CASES, YOU MAY WISH TO BYPASS OUTPUT ENTIRELY, FOR INSTANCE IF YOU KNOW THAT A PARTICULAR COMPILATION WILL NOT PRODUCE VALID RESULTS. USE OF THE SYMBOLIC DSNAME '.D' (OR '.DUMMY'), WILL PREVENT DATA FROM BEING TRANSMITTED TO OR FROM THE COMPUTER.

FURTHER INFORMATION ABOUT DATA SET NAMES MAY BE FOUND IN SECTION 9.A. 'DATA SET NAMING CONVENTIONS.'

(B). USER CATALOGUE, CATPRINT COMMAND

THE NAMES OF ALL CURRENTLY DEFINED DATA SETS OF EACH USER ARE KEPT IN A CATALOGUE KNOWN AS THE LOCATION EQUIVALENCE TABLE (LET). YOU MAY OBTAIN A LISTING OF YOUR CATALOGUE, AS WELL AS THE CATALOGUES OF ALL OTHER USERS WITH DISK CARTRIDGES ON THE SYSTEM AT THE TIME YOU ISSUE THE COMMAND, BY TYPING CATPRINT. IF YOU DESIRE PARTICULAR INFORMATION ABOUT JUST ONE OR A FEW DATA SETS, USE THE DESCRIBE COMMAND.

(C). DATA SET UTILITIES

YOU MAY CREATE, MODIFY, ERASE, AND MOVE DATA SETS USING THE VARIOUS UTILITY COMMANDS SUCH AS COPY, PUNCH, EDIT, ERASE, AND RENAME. ALL SUCH DATA SET MANIPULATION AND UTILITY COMMANDS ARE DESCRIBED IN SECTIONS 7 AND 9 OF THIS MANUAL. EACH COMMAND CAN ALSO BE EXPLAINED BY USING THE COMMAND VERB AS A PARAMETER TO THE EXPLAIN COMMAND (E.G., ==> EXPLAIN COPY).

3. THE SYSTEM EDITOR

THE CYTOS EDITOR IS USED TO CREATE A NEW DATA SET OR TO MODIFY AN EXISTING DATA SET. FOR EXAMPLE, YOU MAY USE THE EDITOR TO CREATE A DATA SET TO BE USED AS INPUT FOR A COMMAND, OR TO CREATE A SOURCE PROGRAM TO BE COMPILED AND RUN FROM YOUR TERMINAL.

COMMAND FORM:

EDIT DSNAME ATTRIBUTE RECORD_LENGTH

DSNAME	NAME OF DATA SET TO BE CREATED OR MODIFIED.
ATTRIBUTE	INDICATES THE TYPE OF DATA SET TO BE EDITED, SO THAT EDITING TABS MAY BE SET. FOR EXAMPLE:

==> EDIT CURVE FORTRAN

INDICATES THAT A FORTRAN SOURCE PROGRAM CALLED CURVE.S IS TO BE CREATED OR MODIFIED. AFTER TYPING A STATEMENT NUMBER, YOU MAY USE THE LOGICALLY DEFINED 'TAB' CHARACTER (INITIALLY '>') FOLLOWED IMMEDIATELY BY THE TEXT OF THE FORTRAN STATEMENT. THE EDITOR WILL AUTOMATICALLY ALIGN THE TEXT IN COLUMN 7. STATEMENTS WITHOUT NUMBERS MAY BE STARTED WITH THE TAB CHARACTER.

RECORD_LENGTH — LENGTH OF RECORDS TO BE CREATED. THE USUAL LENGTH IS 80, TO WHICH THIS PARAMETER DEFAULTS IF YOU SPECIFY NOTHING.

THE EDITOR HAS THREE 'MODES' OF OPERATION: INSERT MODE, EDIT MODE, AND 'Q-MODE'.

IN 'INSERT' MODE (PROMPT: I=>), ALL TEXT TYPED BECOMES PART OF THE DATA SET. TO LEAVE INSERT MODE AND ENTER EDIT MODE, PRESS THE 'INT REQ' KEY.

IN 'EDIT' MODE (PROMPT: E=>) THE EDITOR IS COMMAND-DRIVEN. THE COMMANDS AVAILABLE IN THE EDITOR ARE LISTED UNDER THE EDIT COMMAND IN SECTION 9 OF THIS MANUAL AND ARE EXPLAINED IN DETAIL IN SECTION 5., 'THE EDIT TUTORIAL.'

Q-MODE (PROMPT: Q=>) CAN BE ENTERED BY GIVING THE OPTION Q-MODE ALONG WITH THE ATTRIBUTE PARAMETER WHEN ISSUING THE EDIT COMMAND. THE PURPOSE OF THIS MODE IS TO ALLOW VARIOUS OPTIONS TO BE SET BEFORE YOU ACTUALLY START TO PROVIDE INPUT FOR A NEW DATA SET. YOU MIGHT WISH TO SET TABS, FOR EXAMPLE, OR DEFINE A UTILITY FILE BEFORE ENTERING TEXT IN A DATA SET. SEE THE EDIT TUTORIAL FOR COMMANDS THAT ARE AVAILABLE IN Q-MODE.

4. ANSWERING THE WHATFILE QUESTION

OFTEN A USER COMMAND (ANY PROGRAM COMPILED BY THE USER AND INVOKED USING THE RUN COMMAND) REQUIRES THE USE OF A DISK DATA SET FOR INPUT OR OUTPUT. IN ORDER TO PROVIDE FLEXIBILITY, A PROGRAM REQUIRING THE USE OF A DATA SET REFERS TO IT BY A PSEUDONYM (E.G. THE FORTRAN 'DEFINE FILE' NUMBER). WHEN THE PROGRAM EXECUTES, YOU MAY SPECIFY THE DATA SET YOU WISH TO SUPPLY FOR EACH PSEUDONYM IN THE MAIN PROGRAM. TO DO THIS UNDER THE DISK MONITOR SYSTEM, YOU MUST PROVIDE FILE NAMES (VIA THE '*FILES' CONTROL RECORDS) AT THE TIME YOUR LOAD MODULE (CORE IMAGE PROGRAM) IS CREATED BY THE CORE LOAD BUILDER. UNDER CYTOS, HOWEVER, YOU NEED NOT ESTABLISH THE

RELATIONSHIP BETWEEN THE FILE REFERENCE AND THE ACTUAL DATA SET UNTIL YOU ACTUALLY INTEND TO RUN THE PROGRAM. IN THIS MANNER, YOU MAY REFER TO DIFFERENT DATA SETS EACH TIME YOU INVOKE THE PROGRAM. IF YOU KNOW THAT THE PROGRAM WILL ALWAYS NEED TO USE THE SAME DATA SET, THOUGH, YOU MAY ESTABLISH THE RELATIONSHIP FOR ALL FURTHER EXECUTIONS OF THE PROGRAM BY ISSUING THE NAMEFILE COMMAND.

IF YOU HAVE NOT PREDEFINED THE CORRESPONDENCE BETWEEN THE FILE AND THE DATA SET, YOU WILL BE PROMPTED FOR A DATA SET NAME:

```
WHATFILE 12?>
```

IN MOST CASES A DATA SET NAME WILL SATISFY WHATFILE. IF YOU WISH TO SUPPLY MORE INFORMATION, YOU MAY ENTER ANYTHING IN RESPONSE TO WHATFILE THAT IS A VALID PARAMETER TO THE NAMEFILE COMMAND, EXCEPT THE ACTUAL PSEUDONYM. IN THIS WAY, YOU CAN DEFINE NEW DATA SETS AS THEY ARE NEEDED AND CONTROL THE SIZE IF NEED BE.

FOR EXAMPLE:

```
==> RUN CURVE
WHATFILE 12?> DATA RECORDS 400
```

THE PROGRAM CURVE REQUIRES THE USE OF A DATA SET AND REFERS TO IT WITH THE SYMBOLIC FILE NUMBER 12. THE RESPONSE REQUESTS THE DEFINITION OF A NEW DATA SET CALLED DATA OF SIZE SUFFICIENT TO CONTAIN 400 RECORDS, TO BE USED FOR OUTPUT FROM THE PROGRAM CURVE.

5. ENDING A CYTOS SESSION -- LOGOFF

WHEN YOU HAVE FINISHED ALL YOUR ACTIVITIES USING THE CYTOS SYSTEM, YOU SIGNAL YOUR COMPLETION TO CYTOS BY ISSUING THE LOGOFF COMMAND. AT THIS POINT, CYTOS WILL PROTECT YOUR CATALOGUE AND FILES FROM INADVERTENT USE BY OTHER PEOPLE, AND WILL PREVENT OTHERS FROM ISSUING CYTOS COMMANDS THAT REFER TO YOUR DATA SETS UNLESS YOU HAVE PREVIOUSLY GIVEN PERMISSION TO DO SO.

UNLESS YOU SPECIFIED HOLD AS AN OPTION TO YOUR LOGOFF COMMAND, CYTOS/I CONTROL WILL RETURN TO THE DISK MONITOR SYSTEM IN BATCH MODE. UNDER CYTOS/II, THE TELEPHONE CONNECTION BETWEEN THE COMPUTER AND THE TERMINAL WILL BE BROKEN. IF YOU SPECIFIED HOLD, THE SYSTEM WILL ASK ANOTHER USER TO LOGON, AS DESCRIBED ABOVE IN THE SECTION CALLED 'INVOKING THE CYTOS SYSTEM'. NO OTHER USER WILL BE ALLOWED TO LOGON UNLESS HE HAS MOUNTED HIS OWN LIBRARY (CARTRIDGE) AND SPECIFIES THE CARTRIDGE IDENTIFICATION AS A PARAMETER TO THE COMMAND.

6. SAMPLE SESSSIONS

THE FOLLOWING TEXT CONSISTS OF TWO SAMPLE SESSIONS. THE FIRST ILLUSTRATES THE USE OF AN INTERACTIVE STATISTICAL PROGRAM: THE SECOND SHOWS HOW ONE MIGHT GO ABOUT CREATING, EDITING, COMPILING, AND TESTING A SIMPLE FORTRAN PROGRAM.

```
PLEASE LOGON

==>      LOGON BROWN PRJ15 1005
ENTER PASSWORD>                          (PASSWORD NOT PRINTED BACK)

==> NEWPASS
ENTER NEW PASSWORD >

==>      RGRSN      (REGRESSION PROGRAM INVOKED)

WHATFILE 5?> XDATA                       (XDATA IS THE DISK DATA SET)
>TYPE ? FOR EXPLANATION OTHERWISE 'EOF'>?
COMMAND NAMES: MEAN COR PRINT FILE PLOT DEL ANOVA
OPTIONS: TRAN INT WTS      MATRIX NAMES: X H U INV
*GIVE NOBS,IVAR,DVAR>7 1 1
*GIVE OPTIONS IF ANY>INT
*START COMMANDS

>MEAN
VARIABLE    MEAN     STD DEV

IVAR

         1 30.00 12.00

DVAR
         L 7.080 1.269

>                                        (USER HITS INT REQ, EOF)

==> DESCRIBE XDATA
DATA.SET.NAME---LIB-CART--DBAD--DBCT
XDATA.S          UA  1005  0B40  0120

==> RESUME                               (THE DESCRIBE COMMAND IS ONE
                                         OF SEVERAL THAT MAY BE ISSUED
                                         WHILE A USER COMMAND HAS BEEN
                                         SUSPENDED: RESUME STARTS AGAIN
                                         AT POINT OF SUSPENSION.)
>ANOVA

ANOVA TABLE DEP VAR 1
DUE TO D.O.F. SUM OF SQUARES MEAN SQUARES
REG       1    10.66423        10.66423
DEV       5    0.613774        0.1227543
TOT       6    11.27800
```

```
MULT CORR COEF= 0.97241
STD ERR OF MEAN RES= 0.35036
FRATIO= 86.875       STD ERR OF RES= 0.13242

VAR  REG COEF      STD ERROR      T RATIO ON   5 D.O.F.
  1  0.10286       0.11035E-01    9.3207

                                  (USER HITS 'INT REQ', 'EOF')
==>     LOGOFF




PLEASE LOGON

==>       LOGON BROWN PRJ15 1005
ENTER PASSWORD>

==>       DESCRIBE MYPRG
DATA.SET.NAME---LIB-CART--DBAD--DBCT
MYPRG.S          FX  1005  NOT FOUND (UT01)

==>       EDIT MYPRG FORTRAN        (DATA SET IS NEW SO INPUT
I=>       >DIMENSION A(100)         MODE IS ENTERED DIRECTLY)
I=>       >WRITE (J,1)
I=>       1>FORMAT('ENTER N)
I=>       10>READ (M,2) EN
I=>       >N = EN
I=>       C
I=>       C  NEXT STATEMENT READS UP TO 'N' VALUES FOR 'A'
I=>       C
I=>       READ (M,3) (A(I),I=1,N)
I=>       2>FORMAT(F3.0)
I=>       3>FORMAT(3F10.5)
I=>       >SUM = 0.0
I=>       >DO 20 I=1,N
I=>       20>SUM = SUM + A(I)
I=>       >WRITE (J,4) SUM
I=>       4>FORMAT(F12.6,' IS THE SUM')
I=>       >GO TO 10
I=>       >END
I=>                                 ('INT REQ' TO SWITCH
E=>       TOP,PRINT 20              FROM INPUT TO EDIT MODE)
1+> PRINT 20                        (GENERATED COMMAND)
      DIMENSION A(100)
      WRITE (J,1)
1     FORMAT('ENTER N)
10    READ (M,2) N
      N = EN
```

```
C
C  NEXT STATEMENT READS UP TO 'N' VALUES FOR 'A'
C
READ (M,3) (A(I), I=1,N)
2     FORMAT(F3.0)
3     FORMAT(3F10.5)
      DO 20 I=1,N
20    SUM = SUM + A(I)
      WRITE (J,4) SUM
4     FORMAT(F12.6,' IS THE SUM')
      GO TO 10
      END
--> LAST LINE
E=>    TOP
E=>    AFTER 'ENGER N' ''''        (PLACE A QUOTE AFTER N)
1     FORMAT('ENTER N')
E=>     BEFORE 'READ'#3' '       '    (ADJUST FOR NO TAB ENTERED)
      READ (M,3) (A(I), I=1,N)
E=>     FILE                       (SAVE THE NEW PROGRAM)
MYPRG.S(1005)
CART ID 1005   DB ADDR  3220   DB CNT   0024

==>      FORTRAN MYPRG ADDR (HERE,KEYBOARD,TYPEWRITER)

UNDEFINED VARIABLES
 M         N

COMPILATION DISCONTINUE
ADDR.O(1005)
UT06 ERRORS PREVENT CATALOGUING

==>       EDIT MYPRG FORTRAN
==>       INSERT '*IOCS(TYPEWRITER,KEYBOARD)' (FORTRAN CONTROL CARD
E=>       LOCATE 'DIM'                ALLOWED IN SOURCE PROGRAM)
     DIMENSION A(100)
E=>       INSERT '>M=6'
E=>       INSERT '>N=1'
E=>       FILE NEWPR                 (NEW FILE, BECAUSE NOW LARGER)
NEWPR.S(1005)
CART ID 1005   DB ADDR   3250   DB CNT  002A

==>       ERASE MYPRG                (OLD VERSION NO LOGNER NEEDED)
MYPRG.S(1005)
CART ID 1005   DB ADDR  3220   DB CNT 0024

==>       FORTRAN NEWPR ADDR LIST
ADDR.O(1005)
CART ID 1005   DB ADDR  324A   DB CNT  000E

==>       LOADGO ADDR
ENTER N
5.
6.5       17.204    1.0
42.6      42.7
   110.00400 IS THE SUM
ENTER N
                                  ('INT REQ', 'EOF')
==>       LOGOFF
```

3. INTERACTIVE PROGRAMMING TECHNIQUES

PROGRAMMING FOR AN INTERACTIVE SYSTEM IS SIGNIFICANTLY DIFFERENT FROM PROGRAMMING FOR A BATCH SYSTEM, BECAUSE IN AN INTERACTIVE SYSTEM A HUMAN BEING IS PART OF THE PROCESS. IN BATCH PROGRAMS, MACHINE EFFICIENCY TENDS TO BE OF MAJOR IMPORTANCE; IN INTERACTIVE PROGRAMS, HUMAN FACTORS TEND TO BE OF GREATEST IMPORTANCE.

THE PRIMARY ADVANTAGE OF AN INTERACTIVE SYSTEM IS THAT IT INCLUDES THE USER IN THE PROCESSING; IT ALLOWS ONE TO CHANGE OPTIONS OR PARAMETERS DURING EXECUTION, AND TO DETERMINE IN WHAT DIRECTION PROCESSING SHOULD PROCEED.

BECAUSE A USER IS INVOLVED DIRECTLY, PROGRAMS THAT RUN ON AN INTERACTIVE SYSTEM SHOULD BE DESIGNED TO UTILIZE THIS INVOLVEMENT EFFICIENTLY. COMMUNICATIONS BETWEEN THE PROGRAM AND THE USER BECOME THE MAJOR FACTOR IN THE DESIGN OF AN INTERACTIVE PROGRAM. THE USER MUST ALWAYS BE INFORMED OF WHAT IS HAPPENING, AND IF INPUT IS REQUIRED, HE MUST BE GIVEN ENOUGH INFORMATION TO KNOW WHAT IS NEEDED.

A BASIC TECHNIQUE OF COMMUNICATION IN INTERACTIVE PROGRAMMING IS THE PROMPT. RATHER THAN REQUIRING YOU TO REFER TO A PRINTED MEMO IN ORDER TO PREPARE INPUT, THE INTERACTIVE PROGRAM SHOULD TELL YOU WHAT IS NEEDED AT EACH POINT. A GOOD PROMPT SHOULD COMMUNICATE BRIEFLY AND CLEARLY; A PROMPT WHICH IS TOO VERBOSE IS AS FRUSTRATING AS A PROMPT THAT DOESN'T TELL ENOUGH. THEREFORE, A PROMPT SHOULD BE AS SHORT AS POSSIBLE WHILE STILL GIVING THE NECESSARY INFORMATION.

THE REGRESSION PROGRAM ILLUSTRATED IN THE SAMPLE SESSION IN THE PREVIOUS SECTION IS A GOOD EXAMPLE OF THE USE OF PROMPTS. IT PROVIDES A WAY FOR YOU TO GET A DETAILED EXPLANATION OF HOW THE PROGRAM WORKS IF YOU HAVE NEVER USED IT BEFORE, YET YOU NEED NOT BE ENCUMBERED WITH THIS EXPLANATION IF YOU HAVE PREVIOUSLY EXPERIENCED RUNNING THE PROGRAM. IF YOU CHOOSE TO ANSWER '?' IN ORDER TO GET AN EXPLANATION, YOU WILL RECEIVE ONLY THE MINIMUM INFORMATION NEEDED TO RESPOND TO LATER PROMPTS.

THE EDITOR, SHOWN IN THE SECOND EXAMPLE, ILLUSTRATES A DIFFERENT APPLICATION OF THE "BRIEF BUT CLEAR" PRINCIPLE. THE EDITOR'S PROMPTS ARE DESIGNED FOR THE EXPERIENCED USER, BECAUSE THE EDITOR IS A BASIC TOOL. IF YOU ARE USING THE EDITOR, YOU MUST KNOW WHAT MODE YOU ARE IN, AND WHEN TO ENTER THE NEXT LINE; THE PROMPTS "E=>", "I=>", AND "Q=>" ARE SUFFICIENT.

METHODS OF READING INPUT ARE ALSO AFFECTED BY HUMAN FACTORS. IF YOU ARE SITTING AT THE TERMINAL, YOU WILL NOT FIND IT EASY TO USE THE COLUMN INDICATORS TO GUIDE YOU IN DATA INPUT. THEREFORE, FREE FORMAT FOR DATA INPUT IS DESIRABLE WHEREVER POSSIBLE.

THE PROGRAM CREATED AND TESTED IN THE SECOND SAMPLE SESSION MAKES USE OF THE FREE FORMAT ON INPUT BY USING THE 'FLOATING POINT' FIELD FOR NUMERIC INPUT: EACH NUMBER ENTERED CAN BE ANYWHERE IN THE TEN-COLUMN FIELD (ALTHOUGH EACH FIELD ITSELF MUST BE EXACTLY TEN CHARACTERS LONG, BECAUSE 1130 FORTRAN DOES NOT SUPPORT NON-DISK UNFORMATTED INPUT/OUTPUT OPERATIONS).

4. USER PROGRAMS

WHILE CYTOS PROVIDES A NUMBER OF COMMANDS TO PERFORM FUNCTIONS OF GENERAL USE, A USER MAY WISH TO DEFINE HIS OWN FUNCTION BY WRITING A PROGRAM.

THERE ARE TWO WAYS IN WHICH YOU MAY DEFINE A PROGRAM UNDER CYTOS. YOU MAY CREATE A SOURCE PROGRAM WITH THE EDITOR (OR ENTER A SOURCE PROGRAM INTO YOUR LIBRARY FROM CARDS VIA THE READ COMMAND), COMPILE THE PROGRAM WITH ONE OF THE IBM PROCESSORS, AND EXECUTE THE PROGRAM USING THE "LOADGO" COMMAND; OR YOU MAY TRANSLATE THE PROGRAM WITH A COMPILER, "LINKEDIT" THE PROGRAM INTO A PERMANENT "LOAD MODULE", AND THEN EXECUTE THE PROGRAM EITHER BY GIVING THE "RUN" COMMAND OR BY USING THE NAME OF THE LOAD MODULE AS A CYTOS COMMAND. IN THE LATTER CASE, THE NAME OF YOUR PROGRAM OBVIOUSLY CANNOT BE THE SAME AS A VALID CYTOS SYSTEM COMMAND VERB.

A. CREATING A SOURCE PROGRAM

YOU MAY INITIALLY CREATE A SOURCE PROGRAM EITHER BY USING THE CYTOS EDITOR TO ENTER SOURCE RECORDS INTO A SOURCE DATA SET ON THE DISK, OR BY PUNCHING YOUR SOURCE PROGRAM ONTO CARDS AND READING THE ENTIRE FILE ONTO THE DISK AT ONCE. THE LATTER METHOD IS PREFERABLE FOR LONG PROGRAMS BECAUSE THE SYSTEM IS NOT TIED UP FOR SO LONG A TIME IN THE PROCESS. ONCE THE PROGRAM IS ON DISK, HOWEVER, CONSIDERABLE TIME CAN BE SAVED BY USING THE CYTOS EDITOR TO MAKE ANY NECESSARY CORRECTIONS TO THE SOURCE CODE IN THE PROCESS OF DEBUGGING THE PROGRAM. SINCE UNDER CYTOS ALL THE IBM PROCESSORS ACCEPT INPUT DIRECTLY FROM THE DISK, YOU NEED NOT MAKE CORRECTIONS IN THE ORIGINAL CARD FILE NOR READ THE CARD FILE EACH TIME THE SOURCE PROGRAM IS TO BE COMPILED. ONCE YOUR PROGRAM IS WORKING CORRECTLY, YOU CAN CREATE A NEW CARD FILE FOR BACKUP PURPOSES BY USING THE CYTOS "PUNCH" COMMAND.

IF YOU ARE CREATING THE SOURCE PROGRAM USING THE EDITOR, YOU WILL BE ENTERING SOURCE STATEMENTS IN "INPUT-MODE"; EVERY TIME CYTOS PROMPTS WITH THE CHARACTERS "I=>" YOU MAY ENTER A NEW SOURCE STATEMENT. THE EDITOR'S LOGICAL TAB FACILITY MAKES IT UNNECESSARY FOR YOU TO ALIGN THE VARIOUS FIELDS OF THE SOURCE STATEMENT WITHIN THE INPUT RECORD. EACH TIME YOU USE THE LOGICAL TAB CHARACTER (NORMALLY THE ">"), THE NEXT CHARACTER WILL BE ALIGNED IN THE SAME MANNER AS IF YOU HAD PRESSED THE SKIP KEY ON A STANDARD KEYPUNCH DEVICE. LOGICAL TAB COLUMNS ARE DEFINED FOR ALL THE IBM PROCESSORS (FORTRAN, COBOL, RPG, ASSEMBLER, AND SYSTEM/360 ASSEMBLER FOR PROGRAMS TO BE RUN ON A 360 USING THE REMOTE JOB ENTRY SUBSYSTEM OF THE DISK MONITOR SYSTEM). THE VARIOUS TAB SETTINGS FOR EACH PROCESSOR ARE LISTED UNDER THE "TABS" COMMAND IN THE "EDITOR

TUTORIAL" (SECTION 5 OF THIS MANUAL). YOU MAY USE THE TABS COMMAND TO SET ADDITIONAL OR DIFFERENT TAB COLUMNS, AS NEEDED.

AFTER YOU HAVE ENTERED THE LAST SOURCE STATEMENT, YOU MAY IMMEDIATELY SAVE THE NEW PROGRAM BY PRESSING "INT REQ" TO RETURN TO EDIT-MODE AND THEN ISSUING THE "FILE" COMMAND TO CATALOGUE THE NEW PROGRAM IN YOUR PROGRAM LIBRARY. YOU MAY RE-EDIT THE PROGRAM IMMEDIATELY IF YOU WISH TO CORRECT ANY TYPING ERRORS MADE WHILE ENTERING THE PROGRAM.

B. COMPILING THE SOURCE PROGRAM

BEFORE YOUR PROGRAM CAN BE RUN BY THE 1130 DISK MONITOR SYSTEM IT MUST BE TRANSLATED INTO MACHINE LANGUAGE BY ONE OF THE IBM PROCESSORS. THE LANGUAGE YOU USE TO WRITE YOUR PROGRAM DEPENDS UPON THE PARTICULAR PROBLEM YOU ARE TRYING TO SOLVE; SCIENTIFICALLY-ORIENTED PROBLEMS ARE USUALLY EASIER TO IMPLEMENT USING THE FORTRAN LANGUAGE, WHILE COMMERCIALLY-ORIENTED PROBLEMS CAN USUALLY BE CODED MORE EFFICIENTLY AND UNDERSTANDABLY USING EITHER THE COBOL OR RPG LANGUAGES. NO MATTER WHICH LANGUAGE YOUR PROGRAM IS WRITTEN IN, HOWEVER, AN IBM PROCESSOR MUST BE USED TO TRANSLATE IT INTO THE SET OF MACHINE INSTRUCTIONS BY WHICH THE COMPUTER ACTUALLY SOLVES YOUR PARTICULAR PROBLEM.

TO TRANSLATE THE PROGRAM, ISSUE A CYTOS COMMAND WHICH WILL INVOKE THE APPROPRIATE IBM PROCESSOR AND DIRECT IT TO THE SOURCE PROGRAM YOU HAVE PLACED ON THE DISK. THUS THE "FORTRAN" COMMAND IS USED WHEN YOU WANT TO COMPILE A FORTRAN SOURCE PROGRAM. TYPICALLY, TO COMPILE A PROGRAM, YOU NEED ONLY ISSUE THE COMMAND AND SUPPLY AS PARAMETERS SUFFICIENT INFORMATION TO DESCRIBE WHERE THE SOURCE PROGRAM IS AND WHAT YOU WANT DONE WITH THE "OBJECT PROGRAM" (RESULT OF THE COMPILATION) AFTERWARDS. ADDITIONALLY, YOU MAY SPECIFY THE MANNER IN WHICH YOU WISH TO HAVE THE PROGRAM TRANSLATED, FOR INSTANCE WHETHER OR NOT YOU WANT A LISTING OF THE SOURCE PROGRAM AS PART OF THE PRINTED OUTPUT.

THE VARIOUS COMMANDS YOU MAY USE TO COMPILE A PROGRAM ARE EXPLAINED IN GREATER DETAIL IN SECTION 9.C OF THE MANUAL, "CYTOS COMMANDS".

THE OBJECT PROGRAM PRODUCED BY THE COMPILATION MAY EITHER BE SAVED ON THE DISK AS AN OBJECT MODULE (".O" DATA SET), OR PUNCHED ONTO CARDS IF YOU WISH TO SAVE A COPY FOR BACKUP PURPOSES. BEFORE THE PROGRAM CAN BE RUN (EXEUCTED), HOWEVER, YOU MUST FIRST USE THE IBM CORE LOAD BUILDER TO CREATE A LOAD-MODULE. ALL REFERENCES BY YOUR OWN PROGRAM TO OTHER SUB-PROGRAMS (THOSE YOU MAY HAVE WRITTEN OR SUB-PROGRAMS REFERENCED FROM AMONG THOSE SUPPLIED IN THE IBM PROGRAM LIBRARY) WILL THEREBY BE RESOLVED; A PROGRAM CANNOT BE RUN UNLESS ALL THESE REFERENCED SUB-PROGRAMS ARE AVAILABLE TO YOUR PROGRAM WHEN ACTUALLY CALLED AT EXECUTION TIME.

C. CREATING A LOAD MODULE

IF YOUR PROGRAM IS NOT YET COMPLETELY DEBUGGED, YOU SHOULD INVOKE THE IBM CORE LOAD BUILDER WITH THE LOADGO COMMAND TO CREATE A TEMPORARY LOAD MODULE WHICH WILL IMMEDIATELY BE PLACED INTO EXECUTION. ONCE YOU ARE CONFIDENT THAT YOUR PROGRAM HAS NO MORE ERRORS, YOU MAY CREATE A PERMANENT LOAD MODULE USING THE LINKEDIT COMMAND. THE LOAD MODULE MAY THUS BE CATALOGUED IN YOUR LIBRARY AND SET IN EXECUTION AT ANY TIME BY YOUR ISSUING THE "RUN" COMMAND OR BY USING THE PROGRAM NAME AS A NEW CYTOS COMMAND.

IN SOME INSTANCES THE LOAD MODULE THUS CREATED WILL BE LARGER THAN THE CORE CAPACITY OF THE MACHINE ON WHICH YOUR PROGRAM IS TO BE RUN. THERE ARE TWO SOLUTIONS IN THIS CASE. ONE IS TO BREAK UP YOUR MAIN PROGRAM INTO TWO OR MORE SECTIONS SO THAT ONLY PART OF YOUR PROGRAM IS IN CORE STORAGE AT ANY ONE TIME. EACH SECTION MAY INVOKE THE OTHER BY USING THE LINK FACILITY PROVIDED BY THE DISK MONITOR SYSTEM (SEE THE MANUAL CORRESPONDING TO THE SOURCE LANGUAGE YOU HAVE USED FOR SPECIFIC DETAILS ON HOW ONE PROGRAM MAY LINK TO ANOTHER). ALTERNATIVELY, YOU MAY USE THE OVERLAY FACILITY OF THE CORE LOAD BUILDER TO CONSTRUCT YOUR PROGRAM IN SUCH A MANNER AS TO HAVE IN CORE AT ONE TIME ONLY THOSE SUB-PROGRAMS THAT ARE MOST FREQUENTLY REFERENCED, WHILE THE REMAINING SUB-PROGRAMS CAN BE BROUGHT INTO CORE FROM THE DISK ONLY AS THEY ARE NEEDED. WHILE SUCH A METHOD USUALLY CREATES SLOWER RUNNING PROGRAMS, IT IS OFTEN EASIER AS A WAY OF DECREASING THE EFFECTIVE SIZE OF YOUR LOAD MODULE THAN HAVING TO SPLIT UP YOUR MAIN PROGRAM INTO MULTIPLE SECTIONS. IF YOU INTEND TO USE THIS OVERLAY FEATURE, YOU MUST INDICATE TO THE CORE LOAD BUILDER WITH THE CYTOS "OVERLAY" COMMAND JUST WHICH SUB-PROGRAMS YOU WISH TO HAVE IN CORE ONLY WHEN THEY ARE NEEDED. THE "OVERLAY" COMMAND(S) MUST BE GIVEN PRIOR TO ISSUING THE LOADGO OR LINKEDIT COMMAND THAT WILL CREATE THE FINAL LOAD MODULE.

IF YOU WISH TO INCORPORATE ONE OR MORE SUB-PROGRAMS INTO THE LOAD MODULE FOR TESTING PURPOSES (AS, FOR INSTANCE, TO RESERVE CORE STORAGE FOR THE SUB-PROGRAM EVEN THOUGH IT IS NOT EXPLICITLY REFERENCED BY YOUR OWN MAINLINE PROGRAM AT THE PRESENT TIME), YOU MAY USE THE CYTOS "INCLUDE" COMMAND TO INDICATE TO THE CORE LOAD BUILDER THAT THE SUB-PROGRAM IS TO BE LOADED, EVEN THOUGH IT IS NOT REFERENCED. THIS IS EQUIVALENT TO USING THE *NOCAL FEATURE IN BATCH MODE LOADING OF AN OBJECT MODULE.

YOU MAY ALSO PROVIDE ALTERNATE NAMES TO THE CORE LOAD BUILDER SO THAT WHEN ONE NAME IS REFERENCED BY YOUR OWN PROGRAM, THE CORE LOAD BUILDER WILL USE A DIFFERENT NAME FOR SEARCHING THE SUBROUTINE LIBRARY FOR THE SUB-PROGRAM TO BE INCLUDED IN THE CORE LOAD. THE CYTOS "SUBSTITUTE" COMMAND IS USED TO SPECIFY THESE ALTERNATE NAMES, AND MUST BE ISSUED PRIOR TO THE ACTUAL BUILDING OF THE CORE LOAD.

D. EXECUTING A USER-WRITTEN COMMAND

IF THE CORE LOAD BUILDER IS SUCCESSFUL IN CREATING A LOAD MODULE FROM THE VARIOUS OBJECT MODULES REFERENCED, THE PROGRAM WILL BE READY FOR IMMEDIATE RUNNING. IF YOU HAVE ISSUED THE LOADGO COMMAND, THE PROGRAM WILL BE PLACED INTO EXECUTION AUTOMATICALLY. IF YOU HAVE CREATED AND CATALOGUED A LOAD MODULE USING THE "LINKEDIT" COMMAND, YOU MUST ISSUE THE "RUN" COMMAND TO INITIATE EXECUTION.

YOUR PROGRAM MAY REFER TO ONE OR MORE SYMBOLIC FILES WHOSE CONTENTS ARE TO BE EITHER EXAMINED OR CHANGED AS A RESULT OF THE PROCESS YOU HAVE CODED. WHEN THE LOAD MODULE IS BUILT, THE LOCATION OF THESE FILES IS NOT KNOWN -- THEY MAY NOT IN FACT EXIST AT THE TIME THE LOAD MODULE IS BUILT. BEFORE IT CAN RUN, HOWEVER, YOUR PROGRAM MUST BE PROVIDED SUFFICIENT INFORMATION ON THE WHEREABOUTS AND SIZE OF THE FILES IN ORDER TO ACCESS THEM. FOR EVERY FILE YOUR PROGRAM REFERENCES, YOU WILL BE PROMPTED FOR THE NAME OF A DISK DATA SET WHICH THE PROGRAM CAN USE FOR THIS "RUN"; YOU SHOULD REVIEW SECTION 2.E.4, "ANSWERING THE WHATFILE QUESTION", IF YOU ARE UNFAMILIAR WITH THE MANNER IN WHICH YOU MAY SPECIFY THE REQUIRED INFORMATION. IF YOUR PROGRAM IS ALWAYS GOING TO REFER TO THE SAME SET OF FILES, YOU MAY SPECIFY DEFAULT DATA SET NAMES FOR EACH SYMBOLIC FILE USING THE CYTOS NAMEFILE COMMAND. UNTIL YOU LOGOFF, THE INFORMATION IS SAVED IN THE SYSTEM CONTROL RECORD AREA WHICH IS CONSULTED EVERY TIME YOU INVOKE YOUR PROGRAM FOR EXECUTION. IF A DATA SET NAME HAS PREVIOUSLY BEEN PROVIDED FOR THE SYMBOLIC FILE REFERENCE IN YOUR PROGRAM, YOU WILL NOT BE PROMPTED FOR THE INFORMATION AT EXECUTION TIME. YOU SHOULD TAKE CARE THAT NONE OF YOUR PROGRAMS DESTROYS THE CONTENTS OF A DATA SET REFERRED TO BY AN IDENTICAL SYMBOLIC FILE REFERENCE IN ANOTHER PROGRAM. TO BE SAFE, YOU SHOULD ISSUE THE CLEAR FILES COMMAND WHEN YOU FIRST LOGON TO THE SYSTEM, AND THEN "NAMEFILE" ONLY THOSE FILE REFERENCES WITH UNIQUE FILE PSEUDONYMS.

E. THE DEBUGGING PROCESS

CHANCES ARE YOUR PROGRAM WILL NOT OPERATE COMPLETELY CORRECTLY THE FIRST TIME IT IS EXECUTED. THE ADVANTAGE OF A SYSTEM LIKE CYTOS IS THAT IMMEDIATELY UPON DETECTION OF A DISPARITY BETWEEN ACTUAL AND INTENDED RESULTS, YOU CAN INTERRUPT PROGRAM EXECUTION, GO BACK AND MAKE THE NECESSARY CHANGES IN THE SOURCE PROGRAM OR CORE LOAD BUILDING SPECIFICATIONS, THEN REPEAT THE PROCESS OF BUILDING AND EXECUITNG THE LOAD MODULE. YOU CAN ALSO USE THE VARIOUS DEBUGGING FACILITIES PROVIDED WITH THE IBM PROCESSORS, SUCH AS THE FORTRAN ARITHMETIC TRACE FACILITY, TO DETERMINE JUST WHERE IN YOUR PROGRAM YOUR LOGIC WAS INITIALLY FAULTY.

FOR PARTICULARLY ELUSIVE ERRORS, IT IS SOMETIMES NECESSARY TO EXAMINE THE MACHINE INSTRUCTIONS AND DATA IN CORE AT THE TIME THE

PROGRAM IS RUNNING. YOU MAY USE THE CHECKPOINT COMMAND TO SAVE THE CONTENTS OF CORE STORAGE AS THEY APPEAR AT THE INSTANT YOU PRESS THE "INT REQ" KEY. YOU MAY THEN ISSUE THE CYTOS DUMP COMMAND TO EXAMINE THE CONTENTS OF THE DATA SET CONTAINING THE CHECKPOINTED PROGRAM. THE FIRST 4K OF CORE WILL BE IN THE FIRST 16 SECTORS OF THE DATA SET, THE VERY FIRST WORD OF THE FOURTH SECTOR IN THE DATA SET BEING WORD 6 OF CORE STORAGE. CORE STORAGE FROM WORD 1000 (HEXADECIMAL) TO THE END OF CORE IS POSITIONED STARTING IN SECTOR 17. EVEN THOUGH YOUR PROGRAM HAS BEEN SUSPENDED, YOU CAN USUALLY CONTINUE ITS EXECUTION BY ISSUING THE RESUME OR RESTART COMMANDS. IN THIS MANNER THE PROGRAM MAY BE OBSERVED AT VARIOUS STAGES DURING EXECUTION.

THE PROCESS OF DEGUGGING IS NOT LIMITED TO YOUR PROGRAM AND/OR SUB-PROGRAMS, OF COURSE. OFTEN, ERRORS IN DATA FILES DO NOT SHOW UP UNTIL THEY ARE REFERENCED IN A PARTICULAR MANNER BY A PROGRAM. THE CYTOS EDITOR IS OF VALUE IN CORRECTING DATA ERRORS AS WELL AS IN MODIFYING THE PROGRAMS THAT USE THE DATA FILES.

5. EDITOR TUTORIAL

A. USES OF THE EDITOR

THE CYTOS EDITOR MAY BE USED TO CREATE A NEW DATA SET OR TO MODIFY AN EXISTING DATA SET. ALL DATA TO BE READ WITH A FORMAT SHOULD BE ENTERED THROUGH THE EDITOR TO FACILITATE COLUMN ALIGNMENT; ALL PROGRAMS ENTERED FROM THE USER'S TERMINAL SHOULD BE ENTERED THROUGH THE EDITOR SO THAT TYPING ERRORS MAY BE CORRECTED BEFORE COMPILATION IS ATTEMPTED.

B. COMMAND FORMAT

THE EDITOR IS INVOKED BY ISSUING THE CYTOS COMMAND:

EDIT <DSNAME> <ATR> <RECORD_LENGTH>

THIS COMMAND AND ITS PARAMETERS ARE DESCRIBED IN SECTION 9.C OF THIS REFERENCE MANUAL; THE FOLLOWING TEXT IS A DISCUSSION OF HOW TO USE THE EDITOR ONCE IT HAS BEEN INVOKED.

C. BASIC CONCEPTS

IF THE "DSNAME" PARAMETER IS NULL OR IF IT IS THE NAME OF A DATA SET NOT ALREADY IN THE USER'S CATALOGUE, THE EDITOR WILL ASSUME THAT A NEW DATA SET IS TO BE CREATED, AND IT WILL AUTOMATICALLY ENTER THE INSERT MODE (PROMPT: I=>). IN INSERT MODE ALL TEXT TYPED BECOMES PART OF THE DATA SET. TO LEAVE INSERT MODE AND ENTER EDIT MODE, PRESS THE "INT REQ" KEY.

IF THE DSNAME PARAMETER NAMES AN EXISTING DATA SET, THE EDITOR WILL ASSUME THAT THE DATA SET IS TO BE MODIFIED, AND IT WILL AUTOMATICALLY ENTER THE EDIT MODE (PROMPT: E=>). THE USER MAY THEN ISSUE ANY OF THE EDITOR COMMANDS DESCRIBED IN THIS SECTION. NOTE THAT THESE COMMANDS ARE NOT CYTOS COMMANDS; THEY ARE MEANINGFUL ONLY TO THE EDITOR. TO RETURN TO CYTOS IN ORDER TO ISSUE SYSTEM COMMANDS, ISSUE ONE OF THE EDITOR COMMANDS ATTN, FILE, OR QUIT.

IF THE PARAMATER "QMODE" IS GIVEN AS PART OF THE EDIT COMMAND "ATTRIBUTE" PARAMETER, THE EDITOR WILL AUTOMATICALLY ENTER THE Q-MODE (PROMPT: Q=>), WHETHER THE DATA SET IS NEW OR OLD. THE PURPOSE OF THIS MODE IS TO ALLOW VARIOUS OPTIONS TO BE SET BEFORE INPUT OR EDITING BEGINS. FOR EXAMPLE, ONE MAY USE Q-MODE TO SET TABS OR TO TURN OFF FOLDING (I.E., TO CAUSE THE EDITOR TO DISTINGUISH BETWEEN UPPER AND LOWER CASE). THE COMMANDS AVAILABLE IN Q-MODE ARE THE FOLLOWING:

CWALLS	SET CHANGE WALLS
EXPLAIN	CLARIFY COMMANDS, PARAMETERS
PWALLS	SET PRINT WALLS
TABS	SET TAB COLUMNS
SAVE	SAVE PROCEDURES AND STRING VARIABLES
USE	RESTORE PROCEDURES AND STRING VARIABLES
UTILITY	PROVIDE A WORKFILE FOR THE EDITOR
SET (OR S)	SET LOGICAL CHARACTER FUNCTIONS, OPTIONS
DEFINE	DEFINE A PROCEDURE OF COMMANDS
LABEL	SET RECORD SEQUENCING
QUIT (OR Q)	LEAVE Q-MODE FOR INSERT MODE OR EDIT MODE, AS APPROPRIATE

THE UTILITY COMMAND IS THE ONLY ONE THAT CANNOT ALSO BE ISSUED IN EDIT MODE.

THE CYTOS EDITOR CAN BE USED EITHER AS A CONTEXT EDITOR OR AS A LINE EDITOR. THAT IS, TEXT TO BE EDITED MAY BE SCANNED FOR THE OCCURRENCE OF A PARTICULAR STRING OF CHARACTERS (A "PATTERN"), OR A LINE MAY BE LOCATED BY ITS LINE NUMBER.

THE EDITOR MAINTAINS AN INTERNAL POINTER TO THE "CURRENT LINE". THIS POINTER IS INITIALLY SET AT THE TOP OF THE DATA SET, AT LINE ZERO -- THAT IS, JUST BEFORE THE FIRST LINE. AFTER AN EDIT COMMAND HAS BEEN EXECUTED, THE POINTER IS SET AT THE MOST RECENTLY ACCESSED LINE (WITH TWO EXCEPTIONS). FOR EXAMPLE:

```
==> LOCATE 'ABC'
```

SCANS THE TEXT BEGINNING WITH THE CURRENT LINE FOR THE FIRST LINE CONTAINING THE PATTERN 'ABC'. WHEN A LINE CONTAINING THE PATTERN IS FOUND, THE ENTIRE LINE IS PRINTED, AND IT BECOMES THE CURRENT LINE. THE TWO EXCEPTIONS ARE THE TOP COMMAND, WHICH SETS THE POINTER AT LINE ZERO (BEFORE THE FIRST LINE OF THE DATA SET), AND THE DELETE COMMAND, WHICH SETS THE POINTER AT THE LINE IMMEDIATELY AFTER THE LAST LINE DELETED.

WHEN THE EDITOR IS CREATING A NEW DATA SET (CAUSING INSERT MODE TO BE ENTERED AUTOMATICALLY), INPUT TEXT IS WRITTEN INTO THE DISK DATA SET PERIODICALLY. THIS PROTECTS THE USER FROM LOSING THE INPUT IN CASE OF SYSTEM FAILURE. ONCE EDIT MODE IS ENTERED, HOWEVER, ONLY THE UPDATE AND FILE COMMANDS CAUSE THE MODIFIED TEXT TO BE WRITTEN INTO THE ACTUAL DATA SET. A MODIFICATION TO EXISTING TEXT IS NOT AUTOMATICALLY WRITTEN INTO THE DATA SET NAMED IN THE EDIT COMMAND; UPDATE OR FILE MUST BE EXPLICITLY REQUESTED.

THE USER MAY EXIT FROM THE EDITOR WITHOUT ALTERING THE DATA SET STORED ON THE DISK BY ISSUING THE QUIT COMMAND.

D. EDITOR COMMAND PARAMETERS

COMMANDS TO THE CYTOS EDITOR TAKE THE SAME GENERAL FORM AS CYTOS SYSTEM COMMANDS: A COMMAND VERB IS TYPICALLY FOLLOWED BY ONE OR MORE PARAMETERS. THE NOTATION USED TO DESCRIBE THE EDITOR COMMANDS AND PARAMETERS IS AS FOLLOWS:

COMMAND VERBS MUST BE ENTERED EXACTLY AS GIVEN.	COMMAND
PARAMETERS WHICH NEED NOT BE GIVEN, I.E. ARE OPTIONAL, ARE SURROUNDED BY BRACKETS (THE BRACKETS ARE NOT ENTERED).	COMMAND <OPTIONAL_PARM>
IF A PARAMETER MAY BE A COLLECTION, THIS WILL BE NOTED BY PARENTHESES.	(DSNAME,...)
IF A CHOICE MUST BE MADE BETWEEN TWO OR MORE KEYWORDS, THE POSSIBILITIES WILL BE SEPARATED BY VERTICAL BARS.	OPTA|OPTB|OPTC
A DEFAULT CONDITION WILL BE UNDERLINED.	OPTA|OPTB ----

COMMAND PARAMETERS FALL INTO THE FOLLOWING GENERAL TYPES:

1. DATA SET NAMES - ALL DATA SETS REFERRED TO BY THE EDITOR ARE SOURCE DATA SETS (.S). DATA SETS BELONGING TO OTHER USERS MAY BE EDITED IF PERMISSION TO READ, WRITE, AND COPY HAS BEEN GRANTED BY THE OWNER.

2. OPTIONS - USUALLY "KEYWORDS," ENTERED AS SHOWN.

3. DECIMAL COUNTS - UNSIGNED INTEGERS USED TO SPECIFY DECIMAL QUANTITIES, SUCH AS LOGICAL RECORD LENGTH, LINE NUMBERS, COLUMN SETTINGS, ETC.

4. CHARACTER STRINGS - CHARACTER STRINGS ARE USED FOR SCANNING THE TEXT FOR A SPECIFIC SET OF CHARACTERS ('PATTERNS'), OR FOR INSERTING NEW INFORMATION INTO THE TEXT ('REPLACEMENTS', OR 'LINES'). BY DEFINITION, A CHARACTER STRING IS THE INTERPRETED RESULT OF A SET OF ONE OR MORE CONCATENATED 'CONTEXT ELEMENTS'. THE VARIOUS TYPES OF CONTEXT ELEMENTS THAT MAY BE USED IN THIS MANNER ARE:

A. 'QUOTED STRINGS' - ANY COMBINATION OF LEGAL CHARACTERS ENCLOSED IN SINGLE QUOTES, CALLED A 'QUOTED STRING', IS A CONTEXT ELEMENT. TO REPRESENT A SINGLE QUOTE WITHIN A STRING, USE TWO SINGLE QUOTES IN SUCCESSION. EXAMPLES OF VALID QUOTED STRINGS ARE THE FOLLOWING:

```
'RAILROAD YARD'
'JOHN''S BOOK'
'THE NEXT EXAMPLE IS A NULL STRING'
''
```

A CHARACTER CURRENTLY HAVING THE FUNCTION OF LOGICAL BACKSPACE, ERASE, OR TAB CANNOT BE A CHARACTER IN A STRING. YOU MAY USE THE EDITOR'S SET COMMAND TO ASSIGN THE FUNCTION TO SOME OTHER CHARACTER, AND THEN THE ORIGINAL CHARACTER MAY BE USED IN A STRING.

B. 'NOT' ELEMENT - IF X IS A CONTE
THAT REPRESENTS A SPECIFIC SET OF CHARA
THEN ¬X IS ALSO A CONTEXT ELEMENT. A MATC
WILL OCCUR WHEN THE CHARACTERS IN THE TEXT ARE
NOT THE SAME AS THOSE REPRESENTED BY THE
CONTEXT ELEMENT X. FOR EXAMPLE, THE EDITOR
COMMAND

E=> ANCHOR ¬'000' 72

WOULD CAUSE THE EDITOR TO LOOK FOR THE FIRST
RECORD WITH SOMETHING OTHER THAN '000' IN
COLUMNS 72-74.

THE 'NOT' ELEMENT MAY BE USED ONLY IN SEARCH
PATTERNS.

C. CONCATENATION - IF X AND Y ARE CONTEXT
ELEMENTS, THEN X|Y IS ALSO A CONTEXT ELEMENT.
FOR EXAMPLE,

E=> LOCATE ¬' '|'TEXT'

WOULD SEARCH FOR THE FIRST OCCURRENCE OF THE
STRING 'TEXT' PRECEDED BY A NON-BLANK
CHARACTER.

THE VERTICAL BAR MAY BE OMITTED WHEN THE
CONCATENATION IS CLEAR FROM CONTEXT; FOR
EXAMPLE,

E=> LOCATE 'TEXT'¬' '

WOULD SEARCH FOR THE FIRST OCCURRENCE OF THE
STRING 'TEXT' FOLLOWED BY A NON-BLANK
CHARACTER.

D. STRING VARIABLES - ANY STRING VARIABLE NAME
THAT HAS BEEN DEFINED BY THE 'SET
STRING_VARAIBLE_NAME' COMMAND IS A VALID
CONTEXT ELEMENT. (SEE THE SET COMMAND IN THIS
TUTORIAL.) A STRING VARIABLE NAME MAY BE UP TO
FIVE ALPHABETIC (NOT NUMERIC) CHARACTERS.

R EXAMPLE, ASSUME THAT THE COMMAND

E=> SET NUMS '1234567890'

N ISSUED; THEN

IS 'ABC'NUMS'.'

T TO

34567890.'

STRING VARIABLES MAY BE MODIFIED BY THE 'NOT' ELEMENT, AND/OR BY SUBSTRING NOTATION (DEFINED BELOW).

E. SUBSTRING NOTATION - IF X IS A CONTEXT ELEMENT THAT REPRESENTS A SPECIFIC SET OF CHARACTERS, THEN XM-N, WHERE M AND N ARE INTEGERS SUCH THAT 0<M<N, IS A CONTEXT ELEMENT. XM-N IS THE SUBSET OF X BEGINNING AT THE M'TH CHARACTER AND ENDING WITH THE N'TH CHARACTER. FOR EXAMPLE,

'ABCDEF'2-4

IS EQUIVALENT TO

'BCD'

VAR6-7

REPRESENTS THE SIXTH AND SEVENTH CHARACTERS OF THE STRING VARIABLE VAR.

IF N IS OMITTED (AS IN VAR8), THE SUBSTRING WILL BE THE ONE CHARACTER IN COLUMN M. IF N IS BEYOND THE LAST COLUMN OF THE STRING BEING MODIFIED, IT WILL BE RESET TO THE LAST COLUMN. IF M, THE STARTING VALUE, IS BEYOND THE LAST COLUMN, IT WILL CAUSE AN ERROR.

F. 'CURRENT LINE' ELEMENT - THE LINE CURRENTLY POINTED TO BY THE EDITOR MAY BE REFERRED TO SYMBOLICALLY BY THE CURRENT-LINE CONTEXT ELEMENT, FOR WHICH THE DEFAULT CHARACTER IS THE ASTERISK (SEE THE SET COMMAND). THE CURRENT-LINE FUNCTION IS NOT AVAILABLE ON 4K SYSTEMS.

THE CURRENT-LINE ELEMENT MAY BE MODIFIED BY THE 'NOT' ELEMENT AND/OR BY SUBSTRING NOTATION, FOR EXAMPLE:

```
E=> SET LABL *1-5
E=> NEXT
E=> LOCATE LABL
```

WOULD DEFINE LABL TO BE THE CONTENTS OF THE FORTRAN LABEL FIELD (COLUMNS 1-5) OF THE LINE NOW POINTED TO, AND LOCATE THE NEXT OCCURRENCE OF THAT STRING.

G. 'TAB ELEMENT - ANY OCCURRENCE OF THE CHARACTER CURRENTLY DEFINED AS THE LOGICAL TAB CHARACTER, WHETHER INSIDE OR OUTSIDE A QUOTED STRING, WILL CAUSE THE NEXT CHARACTER IN THE INTERPRETED STRING TO BE PLACED IN COLUMN 'N' OF THE CURRENT RECORD, WHERE 'N' IS THE NEXT HIGHER TAB COLUMN THAN THE COLUMN IN WHICH THE LOGICAL TAB CHARACTER WOULD HAVE FALLEN HAD IT BEEN AN ORDINARY CHARACTER. ALL INTERVENING COLUMNS BETWEEN THE CHARACTER PRECEDING THE LOGICAL TAB AND COLUMN 'N' WILL BE OCCUPIED BY BLANKS. THE FOLLOWING EXAMPLES DEMONSTRATE THE USE OF TAB ELEMENTS:

```
(ASSUME TAB SETTINGS OF 6,11,16):
      COL:      1    6    11   16
                |    |    |    |
'A>B'           A    B
>'C'>'DD>E'          C    DD   E
```

THE TABS COMMAND MAY BE USED TO RESET TAB COLUMN SETTINGS.

H. 'FILLER' ELEMENT - THE 'FILLER' ELEMENT, FOR WHICH THE DEFAULT CHARACTER IS THE POUND SIGN (#), MAY BE USED TO REPRESENT AN UNDEFINED SET OF CHARACTERS IN A SEARCH PATTERN. THE FILLER ELEMENT MAY BE MODIFIED BY A CHARACTER COUNT; OTHERWISE, THE NUMBER OF CHARACTERS IS ALSO UNDEFINED. FOR EXAMPLE,

```
'ABC'#'XYZ'
```

REPRESENTS THE QUOTED STRING 'ABC' FOLLOWED BY ANY SET OF CHARACTERS FOLLOWED BY THE STRING 'XYZ'. IN THE SECOND EXAMPLE,

```
'LDR'#7'PLD'
```

EXACTLY SEVEN CHARACTERS MUST SEPARATE THE QUOTED STRINGS FOR A SUCCESSFUL MATCH.

THE FILLER CHARACTER MAY BE REDEFINED WITH THE SET COMMAND.

THE FOLLOWING TABLE INDICATES WHICH TYPES OF CONTEXT ELEMENTS ARE ALLOWED FOR THE VARIOUS TYPES OF CHARACTER STRINGS USED AS EDIT COMMAND PARAMETERS:

	PATTERN	REPLACE.	LINE	STRING	CHAR
APPLICABLE TO:	LOCATE ANCHOR CHANGE BEFORE AFTER CAPITAL. FOLD	CHANGE BEFORE AFTER OVERLAY	INSERT REPLACE	WHATS SET	SET
ELEMENT TYPE:					
'QUOTE'	YES (¬)	YES	YES	YES	YES
FILLER(S)	YES				
CURRENT-LINE(S)	YES (¬)	YES	YES	YES	
STRING_VAR(S)	YES (¬)	YES	YES	YES	
TAB		YES	YES		

NOTES: (¬) INDICATES 'NOT' ELEMENT ALLOWED
(S) INDICATES SUBSTRING NOTATION ALLOWED

A RATHER COMPLICATED EXAMPLE USING MANY OF THE FEATURES OUTLINED ABOVE IS EXPLAINED BELOW IN DETAIL:

```
E=* CHANGE 'ABC'#3'GHI'#BLANK¬*45-49 'XXX'BLANK'...'
```

SEARCH THE TEXT FOR THE FIRST LINE CONTAINING THE STRING OF CHARACTERS ABC, FOLLOWED BY ANY THREE CHARACTERS, FOLLOWED IMMEDIATELY BY THE CHARACTERS GHI, FOLLOWED BY A STRING OF UNDEFINED LENGTH AND CONTENT, FOLLOWED BY THE ENTIRE STRING SAVED IN THE VARIABLE NAMED BLANK, FOLLOWED BY ANYTHING BUT THE CONTENTS OF COLUMNS 45 THROUGH 49 OF THE RECORD POINTED TO AT THE BEGINNING OF THE SEARCH.

IF THIS STRING IS FOUND, REPLACE IT WITH THE STRING CONSISTING OF THE CHARACTERS XXX, FOLLOWED BY THE CONTENTS OF THE STRING VARIABLE NAMED BLANK, FOLLOWED BY THREE PERIODS. NOTE THAT THE STRING VARIABLE BLANK MAY CONTAIN ANY SET OF CHARACTERS PREVIOUSLY SET.

E. MULTIPLE COMMAND FACILITY

THE CYTOS EDITOR ALLOWS THE USER TO ISSUE MORE THAN ONE COMMAND AFTER ANY PROMPT IN EITHER EDIT MODE OR Q-MODE. NOTE THAT THIS FACILITY IS AVAILABLE ONLY IN THE EDITOR, AND CANNOT BE USED WHILE ISSUING CYTOS SYSTEM COMMANDS. SEVERAL RULES GOVERN THE MANNER IN WHICH MULTIPLE COMMANDS MAY BE ISSUED, VIZ.:

1. COMMANDS MUST BE SEPARATED FROM ONE ANOTHER BY A COMMA.

2. PARAMETERS WITHIN ANY EDITOR COMMAND MUST BE SEPARATED FROM ONE ANOTHER, AS WELL AS FROM THE COMMAND VERB (OR ALIAS), BY A BLANK. HOWEVER, COMMAS MAY BE USED TO SEPARATE ITEMS WITHIN A COLLECTION OF ITEMS REPRESENTING ONE PARAMETER, AS IN THE TABS COMMAND.

3. NO SINGLE COMMAND MAY EXCEED 80 CHARACTERS IN LENGTH. USE COMMAND VERB ALIASES, STRING VARIABLES, AND PROCEDURES WITH SHORT NAMES TO INCREASE THE AMOUNT OF INFORMATION REPRESENTED BY ONE REQUEST.

4. COMMANDS MAY BE CONTINUED ONTO EXTRA LINES BY TYPING THE CONTINUATION CHARACTER AFTER ANY PARAMETER OR AFTER ANY COMMA USED TO SEPARATE COMMANDS FROM ONE ANOTHER; FOR EXAMPLE:

```
E=> TOP, LOCATE 'LEX',&
E&> DELETE 3, PRINT&
E&> 5, TOP
```

5. TOTAL INPUT FOR ANY SINGLE PROMPT IS APPROXIMATELY TWO HUNDRED CHARACTERS (LIMITED TO 80 CHARACTERS ON 4K SYSTEMS).

SEVERAL COMMANDS HAVE AS ONE OF THEIR PARAMETERS A COLLECTION OF ONE OR MORE EDITOR COMMANDS TO BE EXECUTED AT A LATER POINT IN TIME (SEE THE DEFINE AND DO COMMANDS). EDITOR COMMANDS ENTERED IN THIS MANNER ARE SUBJECT TO THE SAME RULES OUTLINED ABOVE. IN ADDITION, THE ENTIRE COLLECTION OF EDITOR COMMANDS MUST BE PLACED IN PARENTHESES:

```
E=> DEFINE MOVE (SET LINE *, DELETE, &
E&> INSERT LINE, PRINT, NEXT), &
E&> LL 32, MOVE
```

THE ABOVE REQUEST WOULD DEFINE A PROCEDURE TO SWITCH THE POSITION OF TWO RECORDS WITHIN THE FILE, THEN ACTUALLY SWITCH THE 32ND AND 33RD RECORDS. MOVE ESSENTIALLY BECOMES ANOTHER EDITOR COMMAND.

F. EDITOR COMMANDS

THE COMMANDS AVAILABLE IN EDIT MODE OR Q-MODE ARE LISTED ALPHABETICALLY IN THE DESCRIPTION OF THE EDIT COMMAND IN SECTION 9. THEY ARE EXPLAINED BELOW IN DETAIL. NOT ALL OF THE FOLLOWING COMMANDS ARE AVAILABLE IN Q-MODE; FOR A LIST OF THOSE THAT ARE, SEE THE DISCUSSION OF Q-MODE UNDER "BASIC CONCEPTS" (SECTION 5.B, ABOVE).

IN THE DESCRIPTIONS THAT FOLLOW, EACH COMMAND VERB IS PRECEDED BY ITS ABBREVIATED ALIAS (IN PARENTHESES), IF ANY, AND FOLLOWED BY ITS PARAMETERS.

```
-----------------------------------------------------------
|                                                         |
|     AFTER  'PATTERN'  'REPLACEMENT'  <NN>  <MM>         |
|                                                         |
-----------------------------------------------------------
```

SEARCH THE TEXT BEGINNING WITH THE CURRENT LINE FOR THE FIRST LINE CONTAINING THE PATTERN STRING STARTING IN OR AFTER COLUMN NN; INSERT THE REPLACEMENT STRING DIRECTLY AFTER THE PATTERN STRING. PERFORM THIS PROCESS MM TIMES, STARTING SUCCESSIVE SEARCHES FOR THE PATTERN IN COLUMN J+1, WHERE COLUMN J IS THE LAST COLUMN OF THE REPLACEMENT STRING JUST INSERTED. IF OMITTED, BOTH NN AND MM DEFAULT TO ONE. THE COMMAND CAUSES THE CURRENT LINE POINTER TO BE SET TO THE CHANGED LINE. THE REPLACEMENT STRING MAY BE OF ANY LENGTH; THE PATTERN STRING MAY BE "NULL", IN WHICH CASE THE EFFECT IS THE SAME AS THE BEFORE COMMAND. IN ANY CASE, THE PORTION OF THE ORIGINAL LINE TO THE RIGHT OF THE PATTERN FOUND WILL BE SHIFTED TO THE RIGHT IN ACCORDANCE WITH THE LIMITATIONS IMPOSED BY ANY "CHANGE WALLS" PREVIOUSLY SET (SEE THE CWALLS COMMAND). FOR EXAMPLE:

```
LINE:      EASYCOME,EASYGO
COMMAND:   E=> AFTER 'EASY' '-' 6
RESULT:    EASYCOME,EASY-GO
```

NOTE THAT THE FIRST OCCURRENCE OF 'EASY' IS BEFORE COLUMN SIX.

```
-----------------------------------------------------------
|                                                         |
|              (A)  ANCHOR  'PATTERN'  <NN>               |
|                                                         |
-----------------------------------------------------------
```

SEARCH THE TEXT BEGINNING WITH THE CURRENT LINE FOR THE FIRST LINE CONTAINING THE PATTERN STRING STARTING EXACTLY IN COLUMN NN (OR IN COLUMN ONE, IF THE SECOND PARAMETER IS OMITTED). FOR EXAMPLE:

```
COMMAND:   E=> ANCHOR 'L' 14
MISSED:    THIS LINE WON'T BE FOUND: IT LACKS AN 'L' IN COL. 14
FOUND:     THIS LINE WILL BE FOUND: IT HAS AN 'L' IN COL. 14
                        |
                COLUMN 14
```

ATTN

CAUSES A CYTOS "ATTENTION". SUSPEND THE EDITOR, ALLOWING SYSTEM COMMANDS TO BE ISSUED. EDITING OF THE CURRENT TEXT MAY BE RESUMED AT THE VERY POINT OF SUSPENSION BY LATER ISSUING THE CYTOS COMMAND RESUME EDIT. IN THE MEANTIME, NO COMMAND MAY BE ISSUED NOR OTHER ACTION TAKEN WHICH WILL DISTURB THE CONTENTS OF THE CORE IMAGE BUFFER (SUCH AS DEPRESSING THE "INT REQ" KEY). FURTHERMORE, IF THE UTILITY EDITOR COMMAND WAS NOT ISSUED AT THE BEGINNING OF THE EDITING SESSION, NO COMMAND SHOULD BE ISSUED THAT WOULD CHANGE THE SIZE OF, OR ALTER THE CONTENTS OF, SYSTEM WORKING STORAGE (E.G., THE "ERASE" COMMAND).

BEFORE 'PATTERN' 'REPLACEMENT' <NN> <MM>

SEARCH THE TEXT BEGINNING WITH THE CURRENT LINE FOR THE FIRST LINE CONTAINING THE PATTERN STRING IN OR AFTER COLUMN NN; INSERT THE REPLACEMENT STRING DIRECTLY BEFORE THE PATTERN STRING, SHIFTING THE PATTERN STRING AND ALL CHARACTERS THEREAFTER TO THE RIGHT J POSITIONS, WHERE J IS THE LENGTH OF THE REPLACEMENT STRING. PERFORM THIS PROCESS MM TIMES, COMMENCING SUCCESSIVE SEARCHES FOR THE PATTERN STRING IN COLUMN K+1, WHERE COLUMN K IS THE LAST COLUMN OF THE PATTERN STRING PREVIOUSLY SHIFTED TO THE RIGHT. THAT PORTION OF THE STRING SO SHIFTED TO THE RIGHT WILL BE TRUNCATED ACCORDING TO THE LIMITATIONS IMPOSED BY ANY CHANGE WALLS PREVIOUSLY SET (SEE THE CWALLS COMMAND). CAUSES THE CURRENT LINE POINTER TO BE SET TO THE CHANGED LINE. IF VERIFICATION IS ON (THE DEFAULT), PRINT THE CHANGED LINE. THE REPLACEMENT STRING MAY BE OF ANY LENGTH; THE PATTERN STRING MAY BE NULL, IN WHICH CASE THE EFFECT IS TO INSERT THE REPLACEMENT STRING AT THE BEGINNING OF THE LINE.

FOR EXAMPLE:

```
LINE:      OVER HILL, OVER DALE, ...
COMMAND:   E=> BEFORE 'OVER' 'HE RAN '  () 2
RESULT:    HE RAN OVER HILL, HE RAN OVER DALE, ...
```

(B) BOTTOM

SET THE CURRENT LINE POINTER AT THE LAST LINE IN THE DATA SET.

```
-----------------------------------------------------------
|                                                         |
|              CAPITALIZE  'PATTERN'  <NN>                |
|                                                         |
-----------------------------------------------------------
```

SEARCH THE TEXT, BEGINNING WITH THE CURRENT LINE, FOR THE FIRST OCCURRENCE OF THE SPECIFIED 'PATTERN' STARTING IN OR AFTER COLUMN NN (COLUMN 1, IF NO COLUMN IS GIVEN). WHEN FOUND, FOLD THE FIRST CHARACTER OF 'PATTERN' TO UPPER CASE. FOLDING SHOULD BE SET OFF WHEN EDITING TEXT CONTAINING LOWER CASE CHARACTERS.

```
-----------------------------------------------------------
|                                                         |
|  (C)  CHANGE  'PATTERN'  'REPLACEMENT'  <NN>  <MM>      |
|                                                         |
-----------------------------------------------------------
```

SEARCH THE TEXT BEGINNING WITH THE CURRENT LINE FOR THE FIRST LINE CONTAINING THE PATTERN STRING, STARTING IN OR AFTER COLUMN NN (COLUMN ONE, IF NN IS OMITTED OR IS A NULL PARAMETER). REPLACE THE PATTERN WITH THE REPLACEMENT STRING, SHIFTING ALL CHARACTERS ORIGINALLY TO THE RIGHT OF THE PATTERN STRING ACCORDING TO THE DIFFERENCE IN LENGTHS BETWEEN THE PATTERN AND REPLACEMENT STRINGS, AND ACCORDING TO ANY LIMITATIONS IMPOSED BY ANY 'CHANGE WALLS' PREVIOUSLY SET. IF MM IS SPECIFIED, REPLACE THE FIRST MM OCCURRENCES OF THE PATTERN STRING IN THE LINE WITH THE REPLACEMENT STRING, OBSERVING IN EACH CASE THE RULES OUTLINED ABOVE. SUCCESSIVE SEARCHES COMMENCE IN COLUMN J+1, WHERE COLUMN J IS THE LAST COLUMN OF THE REPLACEMENT STRING PREVIOUSLY INSERTED.

CAUSES THE CURRENT LINE POINTER TO BE SET TO THE CHANGED LINE. IF VERIFICATION IS ON (THE DEFAULT), PRINT THE CHANGED LINE. THE PATTERN AND REPLACEMENT STRINGS MAY BE OF ANY LENGTH; EITHER MAY BE NULL. FOR EXAMPLE:

```
LINE:       XN PN GN YL HN FD JN
COMMAND:    E=> CHANGE 'N' 'IV' 4 3
RESULT:     XN PIV GIV YL HIV FD JN
```

THE FIRST THREE OCCURRENCES OF 'N' IN OR AFTER COLUMN 4 ARE CHANGED TO 'IV'.

CWALLS (N1 N2 ... N10)

SET COLUMN LIMITS FOR THE AFTER, BEFORE, AND CHANGE COMMANDS. THE PURPOSE IS TO FACILITATE MAINTAINING COLUMN ALIGNMENTS WHILE MAKING CHANGES WITHIN FIELDS.

CONSIDER THE FOLLOWING EXAMPLE OF CHANGES MADE TO TEXT WITHOUT USING THE CWALLS COMMAND:

```
      COLUMN ---- 20          33                      60
                  |           |                       |
                  MACROLIB    (*MACRO UPDATE)         9-03-40
                  NAMEFILE    (FILES)                 9-03-44
                  NEWSPACE    (*STORDATA WS UA)       9-03-46
E=>  PRINT
                  MACROLIB    (*MACRO UPDATE)         9-03-40
E=>  CHANGE 'MACROLIB' 'MACLIB'
                  MACLIB     (*MACRO UPDATE)        9-03-40
E=>  BEFORE 'FILES' '*'
                  NAMEFILE    (*FILES)                 9-03-44
```

CHANGES IN CHARACTER COUNTS HAVE DESTROYED THE COLUMN ALIGNMENTS.

NOW LET US LOOK AT THE SAME EXAMPLE, THIS TIME USING THE CWALLS COMMAND.

ORIGINAL TEXT:

```
                  MACROLIB    (*MACRO UPDATE)         9-03-40
                  NAMEFILE    (FILES)                 9-03-44
                  NEWSAPCE    (*STODATA WS UA)        9-03-46
E=>  CWALLS(20 33 60)
E=>  PRINT
                  MACROLIB    (*MACRO UPDATE)         9-03-40
E=>  CHANGE 'MACROLIB' 'MACLIB'
                  MACLIB      (*MACRO UPDATE)         9-03-40
E=>  BEFORE 'FILES' '*'
                  NAMEFILE    (*FILES)                9-03-44
E=>  AFTER 'STOR' 'E'
                  NEWSPACE    (*STOREDATA WS UA)      9-03-46
```

COLUMN ALIGNMENTS HAVE BEEN MAINTAINED WITHOUT THE USER'S HAVING TO ACCOUNT FOR DIFFERENT WORD LENGTHS.

CWALLS DOES NOT PROTECT ONE AGAINST A REPLACEMENT THAT IS LONGER THAN THE FIELD. IN THAT CASE, THE ORIGINAL FIELD AND THE ONE FOLLOWING IT ARE TREATED AS ONE LONG FIELD.

```
                  NEWSPACE    (*STOREDATA WS UA)      9-03-46
E=>  CHANGE 'NEWSPACE ' 'LONGER THAN FIELD'
                  LONGER THAN FIELD (*STOREDATA WS UA) 9-03-46
```

A CAREFUL DEFINITION OF THE OPERATION OF 'CHANGE WALLS' FOLLOWS:

WHEN THE FIRST OPERAND STRING IS LOCATED IN THE LINE, THE SECOND OPERAND STRING IS INSERTED BEFORE, AFTER, OR DIRECTLY OVER THE FIRST STRING, DEPENDING UPON THE COMMAND. IN ANY CASE, CHARACTERS TO THE RIGHT OF THE FOUND STRING ARE MOVED TO THE LEFT OR RIGHT ACCORDING TO THE DIFFERENCE IN LENGTH BETWEEN THE OLD PATTERN AND THE NEW PATTERN. THE WALL SETTINGS DETERMINE LIMITS FOR THIS MOVEMENT OF RESIDUAL CHARACTERS IN THE LINE. THE RULE IS AS FOLLOWS:

LET COLUMN L BE THE LAST (RIGHTMOST) COLUMN OF THE REPLACEMENT STRING (FOR THE BEFORE COMMAND, THE REPLACEMENT STRING INCLUDES THE PATTERN; THUS COLUMN L IS THE LAST COLUMN OF THE OLD PATTERN). SUPPOSE FURTHER THAT N IS THE FIRST WALL SETTING WHICH IS GREATER THAN L. THEN, AFTER THE SUBSTITUTION HAS TAKEN PLACE, THE LINE FROM COLUMN N TO THE END OF THE RECORD WILL BE AS IT WAS BEFORE THE SEARCH BEGAN. IF THE CHARACTER IN COLUMN N-1 HAS HAD A NET MOVE TO THE LEFT AS THE RESULT OF A CHANGE COMMAND, COLUMNS (L+1) TO (N-1) WILL CONSIST OF THE SUBSTRING S THAT WAS ORIGINALLY TO THE RIGHT OF THE FOUND STRING (UP TO COLUMN (N-1), TOGETHER WITH A NUMBER OF TRAILING BLANKS EQUAL TO THE DIFFERENCE IN LENGTH BETWEEN THE FOUND AND THE SUBSTITUTED STRINGS).

IF THE SUBSTITUTED STRING CAUSES THE CHARACTER IN COLUMN N-1 TO HAVE UNDERTAKEN A NET MOVEMENT TO THE RIGHT (AS IS ALWAYS THE CASE FOR THE AFTER AND BEFORE COMMANDS), THE FIELD FROM COLUMN (L+1) TO COLUMN (N-1) WILL CONTAIN S TRUNCATED ON THE RIGHT BY THE NUMBER OF CHARACTERS EQUAL TO THAT NET MOVEMENT. NOTE THAT IF N=L+1, THE ENTIRE STRING S IS TRUNCATED, AND THE FIELD UNDER CONSIDERATION IS OCCUPIED BY THE SUBSTITUTION (REPLACEMENT) STRING.

```
DEFINE PROCEDURE_NAME (COMMAND,COMMAND,...,COMMAND)
```

DEFINE A SET OF CYTOS EDITOR COMMANDS SUCH THAT, WHEN THE PSEUDO-COMMAND 'PROCEDURE_NAME' IS GIVEN IN RESPONSE TO AN EDITOR PROMPT, THE EFFECT WILL BE THE SAME AS IF A SERIES OF COMMANDS HAD BEEN ISSUED IMMEDIATELY. THE COLLECTION OF EDITOR COMMANDS SAVED IN THIS MANNER MAY INCLUDE ANY LEGAL EDITOR COMMAND (INCLUDING ANY PROCEDURE_NAME AS A COMMAND, OR THE DEFINE COMMAND ITSELF) ALONG WITH THE APPROPRIATE PARAMETERS TO EACH COMMAND.

NOTE THAT STRING_VARIABLE NAMES WILL NOT BE INTERPRETED UNTIL THE COMMAND REFERRING TO THE NAME IS INTERPRETED; CONTENTS OF STRING VARIABLES MAY THUS BE CHANGED DYNAMICALLY WHEN THE PROCEDURE IS INVOKED DIRECTLY, OR INDIRECTLY FROM ANOTHER PROCEDURE.

THE ENTIRE COLLECTION OF COMMANDS TO BE SAVED MAY NOT EXCEED 70 CHARACTERS IN LENGTH. THE EXAMPLE BELOW DEMONSTRATES THE USE OF THE PROCEDURE FACILITY FOR REMOVING THE NEXT OCCURRENCE OF A SPECIFIC STRING OF CHARACTERS FROM A FORTRAN PROGRAM:

```
E=> DEFINE REMOV (LOCATE STR ,CHANGE STR '' 1 80,NEXT)
E=> SET STR 'DIMENSION'
E=> REMOV
2+> LOCATE STR,
      DIMENSION A(100)
2+> CHANGE STR 1 80,
      A(100)
2+> NEXT
E=> SET STR 'FORMAT', REMOV, REMOV
1+> REMOV,
2+> LOCATE STR,
      FORMAT(2I4,6HFORMAT)
2+> CHANGE STR '' 1 80,
      (2I4,6H)
2+> NEXT
1+> REMOV
2+> LOCATE STR,
--> LAST LINE  (ONLY ONE LINE CONTAINING 'FORMAT' IN TEXT.)
2-> CHANGE STR 1 80,
1-> NEXT
E=> ...
```

THE DIGIT IN EACH GENERATED PROMPT INDICATES THE DEPTH OF NESTING OF EACH INTERPRETED COMMAND. THE SIGN TO THE RIGHT OF THE DIGIT INDICATES WHETHER (+) OR NOT (-) THERE ARE MORE LINES IN THE DATA SET AFTER THE CURRENT LINE. IF THE LAST LINE IS REACHED, NO COMMAND THAT MIGHT CAUSE FURTHER CHANGE TO THE TEXT WILL BE ALLOWED UNTIL AN END COMMAND IN THE DEFINITION TURNS OFF THIS WARNING CONDITION.

NOTE: CYTOS EDITOR COMMANDS MAY NOT THEMSELVES BE REDEFINED USING THE DEFINE FACILITY.

(D) DELETE <NN>

DELETE NN LINES, STARTING WITH THE CURRENT LINE. IF THE COUNT IS OMITTED, DELETE ONLY THE CURRENT LINE. CAUSES THE CURRENT LINE POINTER TO BE SET TO THE LINE AFTER THE LAST LINE DELETED.

DISPLAY

DISPLAY THE CURRENT SETTINGS OF ALL LOGICAL CHARACTER FUNCTIONS (SUCH AS THE TAB CHARACTER), AND THE CONTENTS OF ALL CURRENTLY DEFINED PROCEDURES AND STRING_VARIABLES.

DO (COMMAND,COMMAND,...COMMAND) <NN>

DEFINE A TEMPORARY PROCEDURE CONSISTING OF THE SET OF COMMANDS ENCLOSED IN PARENTHESES, THEN SET THE TEMPORARY PROCEDURE INTO EXECUTION. THE PROCEDURE WILL BE EXECUTED NN TIMES (UNLESS THE 'INT REQ' KEY IS PRESSED, OR UNLESS A COMMAND WITHIN THE COLLECTION CAUSES THE LAST LINE TO BE ENCOUNTERED AND THIS CONDITION IS NOT RESET BY THE END COMMAND BEFORE THE LAST COMMAND OF THE TEMPORARY PROCEDURE). IF NN IS OMITTED, THE SET OF COMMANDS WILL BE REPEATED INDEFINITELY (AGAIN, UNLESS INTERRUPTED IN ONE OF THE WAYS DESCRIBED ABOVE). ISSUING A DO COMMAND WITH A COUNT OF ONE IS EQUIVALENT TO ISSUING THE SET OF COMMANDS IMMEDIATELY.

FOR EXAMPLE, TO INSERT A BLANK LINE BEFORE EACH PARAGRAPH OF THE TEXT:

```
E=> TOP, DO (ANCHOR ¬' ' 1,NEXT -1,INSERT ' ',NEXT 2)
```

THE COMMANDS GENERATED FOR THE LAST PARAGRAPH IN THIS PROCESS:

```
...
2+> ANCHOR ' ' 1,
      IN CONCLUSION, LET IT BE FIRMLY STATED
2+> NEXT -1,
2+> INSERT ' ',
2+> NEXT 2
2+> ANCHOR ' ' 1,
--> LAST LINE
2-> NEXT -1,
2-> INSERT ' ',
1-> NEXT 2
E=> ...
```

(E) END

TERMINATE THE BYPASSING OF COMMANDS THAT IS INITIATED WHEN THE SEARCH FOR A PATTERN OF CHARACTERS IS UNSUCCESSFUL AND THE LAST LINE CONDITION IS ENCOUNTERED. THIS COMMAND ALLOWS THE USER TO CONDITIONALLY BYPASS SEVERAL COMMANDS WITHIN A PROCEDURE, DEPENDING UPON WHETHER OR NOT A PARTICULAR STRING IS FOUND:

```
E=> DO (TOP,LOCATE LABL,LL,DELETE,END,SET LABL LABLIX)
```

IN THIS EXAMPLE, THE POINTER IS SET TO THE TOP OF THE DATA SET; SEARCHING COMMENCES FOR THE STRING REPRESENTED BY THE VARIABLE NAME LABL; IF IT IS FOUND IN THE TEST, THE LINE IN WHICH IT IS FIRST FOUND IS PRINTED WITH ITS LINE NUMBER (VIA THE LL COMMAND), THEN DELETED FROM THE TEST. THE VARIABLE STRING IS THEN CONCATENATED TO ANOTHER STRING IN ORDER TO CHANGE THE SEARCH PATTERN, AND THE SEARCH IS INITIATED AGAIN FROM THE TOP OF THE DATA SET. IF ANY PARTICULAR SEARCH IS UNSUCCESSFUL, NOTHING IS DELETED AND ONLY THE VALUE OF THE STRING VARIABLE IS MODIFIED; ALL COMMANDS AFTER THE LOCATE COMMAND AND UP TO THE END COMMAND WILL

EXPLAIN (TERM1 TERM2 ... TERMN)

EXPLAIN EDITOR COMMAND FORMATS, PARAMETERS, AND ERROR MESSAGES. SEE DESCRIPTION OF CYTOS EXPLAIN COMMAND IN SECTION 9.3 FOR FULL DETAILS. THE ALIAS X MAY NOT BE USED FROM WITHIN THE EDITOR.

(F) FILE <DSNAME> <RECORD_LENGTH>

WRITE THE CURRENT VERSION OF THE TEXT INTO THE DATA SET 'DSNAME'; RETURN CONTROL TO CYTOS COMMAND LEVEL. CHANGES MADE TO A DATA SET DO NOT BECOME PERMANENT UNTIL THE DATA SET IS FILED OR UPDATED. IF NO DSNAME IS GIVEN, THE EDITOR ASSUMES THE MOST RECENT DSNAME GIVEN, EITHER IN THE LAST UPDATE COMMAND, OR IN THE EDIT COMMAND ITSELF.

IF 'DSNAME' NAMES A NEW DATA SET, IT WILL BE CREATED WITH THE SPECIFIED RECORD_LENGTH (DEFAULTS TO 80 CHARACTERS PER RECORD), RATHER THAN THAT RECORD_LENGTH ASSOCIATED WITH THE DSNAME GIVEN IN THE EDIT COMMAND. OUTPUT RECORDS WILL BE TRUNCATED OR PADDED WITH BLANKS, AS NECESSARY.

IF 'DSNAME' NAMES AN OLD DATA SET, AND THE CURRENT TEXT IS LARGER THAN THE DATA SET, THE TEXT WILL BE UPDATED INTO THE EDITOR'S WORK FILE, BUT NOT ACTUALLY STORED IN DSNAME.

```
-----------------------------------------------------------
|                                                         |
|                FOLD  'PATTERN'  <NN>                    |
|                                                         |
-----------------------------------------------------------
```

SEARCH THE TEXT, BEGINNING WITH THE CURRENT LINE, FOR THE FIRST OCCURRENCE OF THE 'PATTERN' STARTING IN OR AFTER COLUMN NN (COLUMN 1, IF NO COLUMN IS SPECIFIED). FOLD THE ENTIRE PATTERN TO UPPER CASE. THE ALTERED LINE WILL BE PRINTED IF VERIFICATION HAS PREVIOUSLY BEEN SET ON (THE DEFAULT CONDITION).

```
-----------------------------------------------------------
|                                                         |
|    INFL  DSNAME  <RECORD_LENGTH>  <START>  <END>        |
|                                                         |
-----------------------------------------------------------
```

INSERT THE ENTIRE CONTENTS OF THE DATA SET 'DSNAME' INTO THE TEXT FOLLOWING THE CURRENT LINE. AT THE TIME THAT INFL IS ISSUED, A LINE

INFL ....

IS INSERTED IN THE TEXT. THE ACTUAL COPY OCCURS DURING THE NEXT UPDATE OR FILE. THE RECORD_LENGTH OF THE INSERTED FILE IS ASSUMED TO BE 80 CHARACTERS, UNLESS OTHERWISE SPECIFIED BY THE SECOND PARAMETER.

/START/ IS THE RECORD NUMBER OF THE FIRST RECORD OF THE DATA SET TO BE INSERTED, AND IF OMITTED, DEFAULTS TO ONE. /END/ IS THE RECORD NUMBER OF THE LAST RECORD TO BE INSERTED. IF /END/ IS OMITTED, THE REMAINDER OF THE DATA SET WILL BE INSERTED.

THE INFL LINE WILL BE IMMUNE FROM LOCATION (VIA LOCATE, ETC.) SO AS TO AVOID INADVERTENT ALTERATION OF INTERNALLY-STORED INFORMATION.

(I) INSERT <'LINE'> <NN>

INSERT THE GIVEN LINE AFTER THE CURRENT LINE. IF A COUNT IS GIVEN, INSERT NN COPIES OF THE GIVEN LINE AFTER THE CURRENT LINE. CAUSES THE CURRENT LINE POINTER TO BE SET AT THE LAST LINE INSERTED.

ISSUING THE INSERT COMMAND WITH NO PARAMETERS INDICATES THAT MORE THAN ONE LINE IS TO BE INSERTED AND THAT THE INSERTED LINES ARE TO DIFFER FROM ONE ANOTHER; THE EDITOR WILL ENTER INSERT MODE (PROMPT: I=>), AND ANY TEXT TYPED WILL BECOME PART OF THE DATA SET, UNTIL THE "INT REQ" KEY IS DEPRESSED TO SIGNAL AN "ATTENTION" AND SUBSEQUENT RETURN TO EDIT MODE. IN THE FOLLOWING EXAMPLE THE SYMBOL '%' REPRESENTS PUSHING THE EOF KEY, WHILE THE SYMBOL '!' REPRESENTS AN ATTENTION ("INT REQ" KEY):

```
E=> INSERT 'TWO IDENTICAL LINES' 2,NEXT -1, PRINT 2% !
2+> NEXT -1,
1+> PRINT 2
TWO IDENTICAL LINES
TWO IDENTICAL LINES
E=> INSERT%
I=> THE VERTICAL BAR REPRESENTS THE END OF THE RECORD: % !
I=> THE INPUT WILL BE READ AUTOMATICALLY IF FILLED .....!
I=> OR IF THE EOF KEY IS PRESSED, IN WHICH CASE%
I=> THE RECORD WILL BE PADDED WITH TRAILING BLANKS.%
I=> %
I=> THE PRECEDING LINE WILL BE BLANK.%
I=> IF INPUT RECORDS EXCEED THE LENGTH &%
I&> OF THE CARRIAGE, LOGICAL CONTINUATION CHARACTER '&' &%
I&> MAY BE ENTERED TO CONTINUE THE INPUT FOR&%
I&> THE SAME RECORD ON SUCCESSIVE LINES.%
I=> THE RESPONSE TO THE NEXT PROMPT TERMINATES INSERT MODE.%
I=> !
E=> ...
```

WHETHER IN EDIT MODE OR IN INSERT MODE, ANY OCCURRENCES OF THE PRE-DEFINED LOGICAL 'TAB' CHARACTER WILL HAVE THE EFFECT DESCRIBED IN AN EARLIER SECTION ('TAB' CONTEXT ELEMENTS, SECTION 5.D.4.G). NO OTHER TYPES OF CONTEXT ELEMENTS WILL BE RECOGNIZED IN INSERT MODE, HOWEVER.

```
LABEL  <NN>  <'GANG'>  <NEW>
```

CAUSES LINES TO BE SEQUENCE-NUMBERED DURING EXECUTION OF THE FILE COMMAND. THE FIRST LINE IS NUMBERED NN, AND SUCCEEDING LINES ARE NUMBERED IN INCREMENTS OF NN AND ARE RIGHT-JUSTIFIED WITH LEADING ZEROES. NN MUST BE AN UNSIGNED INTEGER IN THE RANGE 0<NN<32167. 'GANG' REPRESENTS ANY STRING OF UP TO 5 CHARACTERS TO BE GANG-PUNCHED INTO THE LABEL FIELD, LEFT-JUSTIFIED. IF OMITTED, ALL 8 POSITIONS WILL BE NUMERIC.

IF THE NEW OPTION IS SPECIFIED, ONLY THOSE RECORDS ALTERED IN THIS EDITING SESSION WILL BE RE-SEQUENCED.

ISSUING THE LABEL COMMAND WITHOUT A VALUE WILL RESCIND ANY PREVIOUS LABEL COMMAND; THAT IS, LINES WILL NOT BE NUMBERED AT ALL WHEN AND IF A FILE COMMAND IS ISSUED. THE CONTENTS OF THE LAST 8 POSITIONS OF THE RECORD WILL REMAIN UNCHANGED.

```
LL  <NN>
```

LOCATE LINE NN:

SET THE CURRENT LINE POINTER AT THE LINE THAT IS CURRENTLY THE NN'TH LINE IN THE DATA SET, AND PRINT IT. THE LINE NUMBER IS PRINTED WITH THE TEXT OF THE LINE:

```
E=> LL 257
  257:120   FORMAT(/2F3.1,2X,A2)
```

IF NO LINE NUMBER IS GIVEN, THE CURRENT LINE WILL BE PRINTED WITH ITS LINE NUMBER, AND THE CURRENT LINE POINTER WILL REMAIN UNCHANGED.

```
(L)  LOCATE  'PATTERN'  <NN>
```

SEARCH THE TEXT BEGINNING WITH THE CURRENT LINE FOR THE FIRST LINE CONTAINING THE PATTERN STRING STARTING IN OR AFTER COLUMN NN. IF NO COLUMN IS SPECIFIED, COLUMN 1 IS ASSUMED. IF VERIFICATION IS ON (THE DEFAULT), PRINT THE LINE.

```
-----------------------------------------------------------
|                                                         |
|                    (N)  NEXT  <NN>                      |
|                                                         |
-----------------------------------------------------------
```

SET THE CURRENT LINE POINTER NN LINES FROM THE CURRENT LINE. IF NN IS POSITIVE, THE POINTER IS MOVED DOWN NN RECORDS (TOWARDS THE BOTTOM OF THE DATA SET); IF NN IS NEGATIVE, THE POINTER IS MOVED UP NN RECORDS. IF NN IS OMITTED, +1 IS ASSUMED (I.E., POINT TO THE VERY NEXT RECORD).

```
-----------------------------------------------------------
|                                                         |
|          (O)  OVERLAY  'REPLACEMENT'  <NN>              |
|                                                         |
-----------------------------------------------------------
```

OVERLAY THE CURRENT LINE WITH THE REPLACEMENT STRING, STARTING IN COLUMN NN, REPLACING THE OLD CONTENTS FOR AS MANY COLUMNS AS THE LENGTH OF THE REPLACEMENT. THE REST OF THE LINE REMAINS UNCHANGED. IF NN IS OMITTED, COLUMN ONE IS ASSUMED.

```
-----------------------------------------------------------
|                                                         |
|                    PL  <NN>  <MM>                       |
|                                                         |
-----------------------------------------------------------
```

PRINT MM LINES, BEGINNING WITH THE LINE WHICH IS CURRENTLY THE NN'TH LINE OF THE DATA SET. LINE NUMBERS WILL BE PRINTED, AS DESCRIBED FOR THE LL COMMAND. IF NN IS GIVEN AS A NULL PARAMETER (), PRINT MM LINES BEGINNING WITH THE CURRENT LINE. IF MM IS OMITTED, ASSUME ONE LINE. IF BOTH NN AND MM ARE OMITTED, PRINT ONLY THE CURRENT LINE (EQUIVALENT TO ISSUING THE LL COMMAND WITH NO PARAMETERS). FOR EXAMPLE:

```
E=> PL 16 2
   16:      INTEGER B(55),C(55)
   17:10    FORMAT(1H1)

E=> ...
```

(P) PRINT <NN>

PRINT NN LINES, BEGINNING WITH THE CURRENT LINE. PRINTING MAY BE HALTED BY PRESSING "INT REQ"; CONTROL WILL REMAIN IN THE EDITOR. THE CURRENT LINE WILL POINT AT THE LAST LINE PRINTED. IF NN IS OMITTED, PRINT ONLY THE CURRENT LINE.

PWALLS (N1 N2 ... N10)

SET WALLS FOR THE PRINT, LL, AND PL COMMANDS. 'PRINT WALLS' ARE DEFINED AS FOLLOWS:

THE FIRST WALL N1 SPECIFIES THE COLUMN POSITION IN THE LINE TO BE PRINTED WHICH IS TO OCCUPY THE FIRST PRINT POSITION. THE SECOND WALL, N2, SPECIFIES THE COLUMN IN THE LINE TO BE PRINTED WHICH IS TO OCCUPY THE LAST PRINT POSITION.

IF THE FIELD FROM COLUMN N1 TO COLUMN N2 IS BLANK, THE PRINT COMMAND EXAMINES FURTHER WALL SETTINGS N3,N4,...,NK TO DETERMINE WHAT TO DO. IF N2 IS THE HIGHEST SETTING, THEN A BLANK LINE WILL BE PRINTED. OTHERWISE, PRINT WILL LOOK FOR NON-BLANKS BETWEEN COLUMNS N2 AND N3; IF IT FINDS ONE, THE FIELD FROM COLUMN N1 TO COLUMN N3 WILL BE PRINTED.

IN GENERAL, IF THE FIELD FROM COLUMN N1 TO COLUMN NK IS BLANK, AND THERE IS NO WALL SET HIGHER THAN NK, A BLANK LINE WILL BE PRINTED. OTHERWISE, IF N(K+1) IS THE NEXT HIGHER SETTING, AND A NON-BLANK OCCURS BETWEEN COLUMNS NK AND N(K+1), THE PRINTED LINE WILL CONSIST OF THE FIELD FROM COLUMN N1 TO COLUMN N(K+1).

(Q) QUIT

IF ISSUED IN EDIT MODE, LEAVE THE EDITOR AND RETURN TO CYTOS COMMAND LEVEL WITHOUT FILING THE CURRENT TEXT.

IF ISSUED IN Q-MODE, LEAVE Q-MODE AND ENTER INSERT OR EDIT MODE, DEPENDING ON WHETHER THE DSNAME PARAMETER TO THE EDIT COMMAND NAMED A NEW OR OLD DATA SET.

```
(R) REPLACE 'LINE' <NN>
```

REPLACE THE CURRENT LINE WITH THE GIVEN LINE. IF A COUNT IS GIVEN, REPLACE THE NEXT NN LINES WITH THE GIVEN LINE, STARTING WITH THE CURRENT LINE. THE 'CURRENT LINE' IS SET TO THE LAST LINE REPLACED.

```
SAVE DSNAME
```

SAVE ALL CURRENT PROCEDURE AND STRING_VARIABLE DEFINITIONS IN THE OLD (EXISTING) DATA SET 'DSNAME' (THE SIZE OF WHICH MUST BE AT LEAST 26 'CARDS' LONG). THE USE COMMAND MAY BE USED DURING A LATER EDITING SESSION TO RESTORE THE SAME SET OF PROCEDURES, STRING_VARIABLES, WALL AND TAB SETTINGS.

```
                 1. STRING_VARIABLE_NAME   <'STRING'>
                 2. VERIFICATION|EXPANSION|FOLDING  <ON|OFF>
(S) SET          3. BACKSPACE|ERASE|CONCATENATION|
                    TAB|CONTINUATION|FILLER|NOT|          'CHAR'
                    CURRENT LINE|QUOTE
```

SET VARIOUS VALUES WITHIN THE EDITOR.

(1). THE PRIMARY USE OF THE SET COMMAND IS TO EQUATE A STRING_VARIABLE_NAME WITH AN ACTUAL STRING OF CHARACTERS, SO THAT THE NAME MAY BE USED IN SUBSEQUENT COMMANDS AS A 'CONTEXT ELEMENT'. FOR EXAMPLE:

```
E=> SET DIGIT '1234567890',SET C DIGIT,SET C CICICICICICICIC
2+> SET C DIGIT,
2+> SET C CICICICICICICIC
E=> WHATS C
12345678901234567890123456789012 3...                    ...7890
|                                                             |
COL. 1                                                   COL. 80
```

OMISSION OF THE SECOND PARAMETER WILL DISSOCIATE THE STRING_VARIABLE_NAME FROM ANY STRING. A MAXIMUM OF 12 VARIABLES MAY BE IN USE AT ANY GIVEN TIME. OPTIONS TO THE OTHER FORMS OF THE SET COMMAND (E.G. TAB, VERIFICATION) MAY NOT BE USED AS STRING_VARIABLE_NAMES. STRING_VARIABLE_NAMES MAY CONSIST OF UP TO FIVE (5) ALPHABETIC (NOT NUMERIC) CHARACTERS.

(2). SET INTERNAL SWITCHES ON (THE DEFAULT), OR OFF. ALL SWITCHES ARE INITIALLY SET ON.

A. VERIFICATION CONTROLS THE PRINTING OF LINES AFTER SUCCESSFUL SEARCHES FOR PATTERN STRINGS, SUCH AS BY THE LOCATE AND CHANGE COMMANDS.

B. EXPANSION CONTROLS THE PRINTING OF COMMANDS GENERATED AS A RESULT OF ISSUING PROCEDURE_NAMES AS COMMANDS, OR MULTIPLE COMMANDS (I.E. ALL COMMANDS WITH THE PSEUDO-PROMPT FORM N+> AND N->, WHERE N IS THE CURRENT PROCEDURE NESTING DEPTH).

C. FOLDING CAUSES THE EDITOR TO TRANSLATE ANY AND ALL LOWER CASE INPUT CHARACTERS INTO UPPER CASE BEFORE ANY TEXTUAL SEARCHES OR COMPARISONS ARE MADE. (ON THE 1130, LOWER CASE CHARACTERS MAY NOT BE ENTERED FROM THE CONSOLE KEYBOARD.)

(3). SET LOGICAL CHARACTER FUNCTIONS. VARIOUS FUNCTIONS ARE PERFORMED BY SPECIFIC TYPED CHARACTERS IN CERTAIN EDITING CONTEXTS. IF THE DEFAULT CHARACTERS ARE NOT CONVENIENT TO THE USER, HE MAY RESET ANY CHARACTER FUNCTION WITH ANOTHER CHARACTER USING THE SET COMMAND. THE DEFAULT SETTINGS ARE:

OPTION	CHARACTER	FUNCTION
BACKSPACE	@ (AT-SIGN)	RUB-OUT CHARACTER
CONCATENATION	| (VERTICAL BAR)	CONCATENATE CONTEXT ELEMENTS
CONTINUATION	& (AMPERSAND)	CARRIAGE RETURN FOR INPUT
CURRENT_LINE	* (ASTERISK)	CONTEXT ELEMENT
ERASE	¢ (CENT-SIGN)	LOGICAL 'ERASE FIELD' KEY
FILLER	# (NUMBER-SIGN)	'UNDEFINED' CONTEXT ELEMENT
NOT	* (LOGICAL NOT)	'ANYTHING BUT' MODIFIER
QUOTE	' (PRIME)	'QUOTED STRING' DELIMITER
TAB	> (RIGHT ARROW)	TAB RIGHTWARDS

ANY CHARACTER ON A GREY KEY EXCEPT THE BACKSPACE KEY (<--) MAY BE DEFINED AS A LOGICAL CHARACTER FUNCTION. A FUNCTION MAY NOT BE RESET USING ANY CHARACTER ALREADY IN USE BY ANOTHER FUNCTION. ONLY THE FIRST FIVE CHARACTERS OF THE OPTION NEED BE GIVEN. FOR EXAMPLE:

```
E=> SET BACKS '%'
E=> INSERT 'IN ORDER TO USE THE CHARACTER ''@'' IN TEXT,'
E=> INSERT 'YOU MUST FIRST SET THE BACKSPACE FUNCTION TO'
E=> INSO%ERT 'SOME OTHER CHARACTER.'
E=> NEXT -2, PRINT 3
IN ORDER TO USE THE CHARACTER '@' IN TEXT,
YOU MUST FIRST SET THE BACKSPACE FUNCTION TO
SOME OTHER CHARACTER.
```

TABS (N1 N2 ... N10)

SET LOGICAL TABS AT COLUMNS N1, N2, N3, ... UP TO N10 (0<N1<N2<...<N10<RECORD_LENGTH). THE TABS ARE LOGICAL, NOT PHYSICAL. THAT IS, THEY DO NOT NECESSARILY CORRESPOND TO THE TAB POSITIONS SET AT THE TERMINAL.

A TAB WILL OCCUR ANY TIME THE LOGICAL TAB CHARACTER (DEFAULT: '>') IS USED IN OR BETWEEN CONTEXT ELEMENTS IN THE INSERT, REPLACE, OR OVERLAY COMMANDS, OR IS IMBEDDED IN INSERT MODE INPUT.

OPTIONS TO THE EDIT COMMAND SET INITIAL TABS IN THE FOLLOWING MANNER:

FORTRAN	7, 12, 17, 73
ASSEMBLER	21, 27, 32, 33, 35, 45, 73
COBOL	8, 12, 16, 28, 36, 44, 73
RPG	73
ASM360	10, 16, 36, 73
DATA	11, 21, 31, 41, 51, 61, 71

(T) TOP

SET THE CURRENT LINE POINTER AT THE TOP OF THE DATA SET (AT LINE ZERO, BEFORE THE FIRST LINE OF THE DATA SET).

(U) UPDATE <DSNAME> <RECORD_LENGTH>

WRITE THE CURRENT VERSION OF THE TEXT INTO THE OLD DATA SET 'DSNAME', USING THE SPECIFIED RECORD_LENGTH (DEFAULTS TO 80 CHARACTERS, IF OMITTED). RETURN CONTROL TO THE EDITOR RATHER THAN TO CYTOS COMMAND LEVEL. IF NO DSNAME IS GIVEN, THE EDITOR ASSUMES THE MOST RECENT DSNAME GIVEN, EITHER IN THE LAST UPDATE COMMAND, OR IN THE EDIT COMMAND ITSELF. (IF EDITING A NEW DATA SET, THE TEXT IS WRITTEN TO SYSTEM WORKING STORAGE.)

USE DSNAME

RESTORE DEFINITIONS OF PROCEDURES AND STRING_VARIABLES PREVIOUSLY SAVED IN THE DATA SET DSNAME. SAVE AND USE ALLOW YOU TO SAVE THIS INFORMATION BETWEEN EDITING SESSIONS.

UTILITY DSNAME

THE UTILITY COMMAND CAN BE ISSUED ONLY IN Q-MODE. IT CAUSES THE DATA SET 'DSNAME' TO BE USED AS THE EDITOR'S UTILITY FILE. THERE ARE THREE POSSIBLE ADVANTAGES TO BE GAINED FROM ISSUING THIS COMMAND: (A) IN THE EVENT OF SYSTEM FAILURE (IN CYTOS OR IN EDIT), THE PROBABILITY OF SAVING AN EDITED TEXT IS SOMEWHAT HIGHER; (B) EDITING SPEED IS INCREASED IN THE CASE THAT THE INPUT DATA SET AND SYSTEM WORKING STORAGE ARE WIDELY SEPARATED ON THE SAME DISK CARTRIDGE; THE USER MAY ASSIGN A DATA SET CLOSER TO THE INPUT FILE OR ON A CARTRIDGE DIFFERENT FROM THAT OF THE INPUT FILE, IN ORDER TO MINIMIZE DISK ARM MOVEMENT DURING UPDATING OR SEARCHING OF THE TEXT; (C) THE USER MAY IGNORE THE RESTRICTIONS REGARDING SYSTEM WORKING STORAGE UPON ISSUING THE ATTN COMMAND, SINCE WORKING STORAGE IS NOT USED IF A UTILITY DATA SET IS SUPPLIED.

(W) WHATS ELEMENT

PRINT THE CURRENT CONTENTS OF VARIOUS TYPES OF CONTEXT ELEMENTS, BUT MOST USEFULLY, OF STRING_VARIABLES. FOR EXAMPLE:

```
E=> WHATS DIGIT'...'DIGIT
1234567890...1234567890
```

'PROCEDURE_NAME'

ANY DEFINED PROCEDURE MAY BE USED AS AN EDITOR COMMAND. SEE THE DESCRIPTION OF THE DEFINE COMMAND FOR FURTHER DETAILS.

6. PERMISSION TUTORIAL

THE ABILITY TO SHARE A DATA SET (CONTAINING EITHER A PROGRAM OR DATA) WITH OTHER PEOPLE USING THE SYSTEM IS ONE OF THE MOST USEFUL FEATURES OF A SYSTEM LIKE CYTOS. ON THE OTHER HAND, FILES STORED ON DISK CAN EASILY BE ERASED AND MUST BE PROTECTED FROM INADVERTENT DESTRUCTION. CYTOS ALLOWS YOU TO SHARE YOUR DATA SETS WITH OTHER USERS WHILE PROVIDING A REASONABLY SECURE SYSTEM OF PROTECTING YOUR OWN DATA SETS FROM OTHERS WHILE THE SYSTEM IS NOT UNDER YOUR CONTROL.

YOU CAN OF COURSE ENTIRELY REMOVE YOUR LIBRARIES FROM THE SYSTEM BY DISMOUNTING THEM AFTER YOU HAVE LOGGED OFF. THIS IS THE SUREST FORM OF PROTECTION. HOWEVER, IF YOU WANT TO SHARE YOUR DATA SETS, YOU MAY PROVIDE ACCESS TO THEM FOR VARIOUS PURPOSES AND TO VARIOUS CLASSES OF USERS.

THE 'AUTHOR' OF A DATA SET IS THE PERSON WHO CREATES IT. ONLY THE AUTHOR CAN DECIDE WHAT DEGREE OF PROTECTION THE DATA SET IS TO HAVE AND WHO IS ALLOWED TO USE IT. THE AUTHOR MAY DETERMINE WHO SHALL USE HIS DATA SETS AND ALSO WHAT THEY MAY DO WITH THEM. FOR EXAMPLE, HE MAY DECIDE THAT ALL USERS ON HIS PROJECT MAY READ HIS DATA SETS, BUT NOT WRITE INTO THEM. HE MAY DECIDE TO PROTECT HIS DATA SETS FROM HIMSELF, SO THAT HE WILL BE WARNED IF HE SHOULD TRY TO ERASE THEM INADVERTENTLY.

THERE ARE THREE DIFFERENT CLASSES OF SHARING IN THE CYTOS SYSTEM. THE AUTHOR MAY ALLOW ACCESS TO HIS DATA SETS TO:

1. HIMSELF (AUTHOR);
2. ALL USERS ON HIS PROJECT; OR
3. ALL OTHER USERS OF THE CYTOS SYSTEM.

IN ADDITION, THERE ARE FOUR TYPES OF ACCESS THAT CAN BE GRANTED. THEY ARE:

1. READ - IF THE DATA SET IS A PROGRAM, IT MAY BE RUN; IF IT IS A SOURCE DATA SET, IT MAY BE USED AS INPUT TO A COMMAND OR IT MAY BE PRINTED AT THE TERMINAL;

2. WRITE - THE DATA SETS MAY BE WRITTEN INTO;

3. COPY - THE DATA SETS MAY BE COPIED USING THE COPY COMMAND; AND

4. ERASE - THE DATA SETS MAY BE ERASED USING THE ERASE COMMAND.

TO GRANT PERMISSION TO ACCESS, USE THE PERMIT COMMAND; TO REVOKE PERMISSION, USE THE UNPERMIT COMMAND; TO PRINT THE CURRENT PERMISSION STATUS, USE EITHER COMMAND WITHOUT ANY PARAMETERS.

EXAMPLES:

1. AFTER YOU HAVE ESTABLISHED OWNERSHIP OF A VOLUME, ALL DATA SETS ON THAT VOLUME WILL AUTOMATICALLY HAVE THE FOLLOWING STATUS:

```
    AUTHOR     R W C E
    PROJECT
    OTHER
***END OF PERMIT LIST***
```

TO ISSUE PERMISSION FOR ALL USERS ON THE PROJECT TO READ YOUR DATA SETS.

```
==> PERMIT 1005 READ PROJECT
    AUTHOR     R W C E
    PROJECT    R
    OTHER
***END OF PERMIT LIST***
```

2. SUPPOSE THAT THE STATUS OF PERMISSION FOR VOLUME 2626 IS AS FOLLOWS:

```
    AUTHOR     R W C E
    PROJECT    R   C
    OTHER      R   C
***END OF PERMIT LIST***
```

THEN, TO REVOKE PERMISSION TO READ FROM USERS NOT ON THE PROJECT.

```
==> UNPERMIT

    AUTHOR     R W C E
    PROJECT    R   C
    OTHER          C
***END OF PERMIT LIST***
```

NOTE: WHEN PERMITTING DATA SETS TO BE USED IN RESPONSE TO THE WHATFILE QUESTION, YOU MUST GIVE BOTH READ AND WRITE PERMISSION, EVEN THOUGH A PARTICULAR DATA SET WILL ONLY BE USED FOR ONE OR THE OTHER (I.E. EITHER FOR INPUT OR OUTPUT), BECAUSE WHEN THE DATA SETS ARE OPENED, IT IS NOT POSSIBLE TO TELL HOW THEY WILL BE USED BY THE EXECUTING PROGRAM.

7. UTILITY TUTORIAL

CYTOS HAS A FULL COMPLEMENT OF UTILITY COMMANDS TO ASSIST YOU IN MAINTAINING YOUR PROGRAM AND DATA LIBRARY. THESE COMMANDS ARE DESIGNED TO CREATE, MOVE, AND OTHERWISE MANIPULATE ENTIRE DATA SETS WITH THE MINIMUM CONCERN ON YOUR PART FOR JUST WHAT TYPE OF INFORMATION IS CONTAINED IN THE DATA SET BEING DEALT WITH, OR HOW LARGE THE DATA SET IS AT THE TIME YOU WANT TO DO SOMETHING WITH IT.

ALL SOURCE DATA SETS (I.E., THOSE WHOSE TYPE IS .S) MAY BE:

1. PRINTED EITHER ON THE SYSTEM PRINTER, OR AT THE TERMINAL (BY SPECIFYING "HERE");

2. PUNCHED USING THE SYSTEM OUTPUT DEVICE (EITHER A CARD OR PAPERTAPE PUNCH, DEPENDING UPON YOUR SYSTEM);

3. CREATED, BY READING THE SOURCE FROM CARDS VIA THE SYSTEM INPUT DEVICE;

4. ERASED, OR REMOVED FROM THE LIBRARY ENTIRELY;

5. RENAMED TO ANOTHER SIMPLE DSNAME (THE USERID AND THE TYPE REMAIN THE SAME, HOWEVER);

6. COPIED, USING THE "COPY" COMMAND;

7. ALLOCATED SPACE ON A DIRECT-ACCESS VOLUME BUT NOT FILLED WITH ANYTHING (VIA THE NEWSPACE OR ALLOCATE COMMAND);

8. AND LISTED OR REPRODUCED, IF THE SOURCE IS ON CARDS IN THE FIRST PLACE.

IN ADDITION, THE FOLLOWING COMMANDS MAY BE USED FOR ALL TYPES OF DATA SETS:

1. "REMOVE" WILL COPY THE DATA SET TO CARDS (BINARY FORMAT) AND WILL, OPTIONALLY, ERASE THE DATA SET FROM THE DISK;

2. "RESTORE" WILL REVERSE THE PROCESS, I.E., CREATE A DISK COPY OF A BINARY CARD FILE (OF ANY TYPE);

3. "ERASE" IS APPLICABLE TO ANY TYPE OF DATA SET;

4. "CATALOGUE" WILL MOVE A DATA SET FROM THE SYSTEM 'WORKING STORAGE' TO THE USER AREA (YOUR LIBRARY) AND MAKE AN ENTRY FOR IT IN THE LOCATION EQUIVALENCE TABLE (LET);

5. "UNCATALOGUE" WILL MOVE A DATA SET FROM YOUR LIBRARY TO SYSTEM WORKING STORAGE, THEN REMOVE THE REFERENCE TO IT FROM LET:

6. "RECATALOGUE" WILL MOVE A DATA SET FROM SYSTEM WORKING STORAGE INTO AN OLD (EXISTING) DATA SET, AS LONG AS THE NEW VERSION IS NO BIGGER THAN THE OLD; THE CATALOGUE ENTRY IN LET REMAINS UNCHANGED.

7. "WORKON" WILL COPY A DATA SET FROM YOUR LIBRARY TO SYSTEM WORKING STORAGE, BUT WILL NOT REMOVE THE LET ENTRY.

THE LAST FOUR COMMANDS ARE PRIMARILY OF USE FOR RECOVERING FROM SITUATIONS IN WHICH A DATA SET MAY NOT HAVE BEEN MOVED, CREATED, OR WHATEVER BECAUSE OF AN ERROR IN COMPILATION OR SOME OTHER PREVIOUS FUNCTION. IF YOU WANT TO SAVE THE DATA SET AND GIVE IT A NAME ANYWAY, THESE COMMANDS WILL BE OF SOME USE.

ALL OF THE COMMANDS MENTIONED IN THIS TUTORIAL, ALONG WITH THE APPROPRIATE PARAMETERS TO EACH, ARE EXPLAINED IN DETAIL IN SECTION 9 OF THIS REFERENCE MANUAL.

RULE OF THUMB: WHEN IN DOUBT, COPY FIRST, THEN FIDDLE ...

---

8. CARD-INPUT MODE

THE USER OF A FACILITY ON WHICH AN INTERACTIVE SYSTEM CO-EXISTS WITH A BATCH OPERATING SYSTEM HAS THE FLEXIBILITY OF USING EITHER OR BOTH TO SOLVE A DATA PROCESSING PROBLEM. CERTAIN APPLICATIONS LEND THEMSELVES TO THE INTERACTIVE CAPABILITY BECAUSE OF THE EASE WITH WHICH THE DIRECTION OF PROCESSING CAN BE CONTROLLED AT THE TIME THE PROBLEM IS BEING SOLVED; OTHERS WILL REQUIRE PROCESSING OF LARGE AMOUNTS OF DATA, WITHOUT HUMAN INTERVENTION, ALLOWING EFFICIENT USE OF THE BATCH FACILITIES. ALTHOUGH DESIGNED PRIMARILY TO DIRECT PROCESSING IN A BATCH ENVIRONMENT. THIS SECTION EXPLAINS HOW TO USE THE CARD-INPUT MODE OF THE CYTOS SYSTEM AS AN ADJUNCT AND/OR ALTERNATIVE TO THE JOB CONTROL FEATURES OF THE STANDARD DISK MONITOR SYSTEM.

A. THE CONCEPT OF A CYTOS JOB

THE MAJOR UNIT OF PROCESSING UNDER THE DISK MONITOR SYSTEM IS THE 'JOB', INITIATED BY THE OCCURRENCE OF A '// JOB' CONTROL RECORD IN THE INPUT STREAM. MINOR UNITS OF PROCESSING, WHICH WE SHALL CALL 'TASKS', ARE GROUPED TO FORM A MAJOR JOB. EACH TASK IS ALSO PRECEDED BY A MONITOR CONTROL CARD, E.G., '// FOR' AND '// XEQ'. IN THIS SENSE, THE OCCURRENCE OF A '// CYTOS' CONTROL RECORD IN THE INPUT STREAM DEFINES A DMS TASK.

HOWEVER, BECAUSE ALL FACILITIES OF THE DISK MONITOR SYSTEM MAY BE USED UNDER CYTOS, RUNNING CYTOS TAKES THE FORM OF A 'SUPER-TASK;' THE CYTOS MONITOR IS IN CONTROL OF ALL PROCESSING AND THE DISK MONITOR SYSTEM REMAINS DORMAT UNTIL THE USER ISSUES THE "LOGOFF" COMMAND. FOR THIS REASON, WE MAY SPEAK OF A CYTOS SESSION AS A CYTOS 'JOB'. A MINOR UNIT OF PROCESSING (CYTOS 'TASK') IS INITIATED BY ISSUING A CYTOS COMMAND. BECAUSE CYTOS COMMANDS MAY BE GROUPED AND ENTERED IN CARD FORM, WHAT WE HAVE IS AN ALTERNATE METHOD OF BATCH PROCESSING. A CYTOS JOB CONSISTS OF THOSE CYTOS COMMANDS REQUIRED TO SOLVE A DATA PROCESSING PROBLEM IN BATCH MODE.

FROM THE POINT OF VIEW OF THE CYTOS SYSTEM, THEN, COMMANDS CAN BE INTERPRETED IN ONE OF SEVERAL MODES:

CONVERSATIONAL MODE, IN WHICH COMMANDS ARE ACCEPTED FROM THE TERMINAL, AND UNDER WHICH THE USER IS FREE TO EXPLOIT THE INTERACTIVE CAPABILITIES OF THE SYSTEM;

CARD-INPUT MODE, IN WHICH COMMANDS ARE GROUPED INTO A CYTOS JOB TO SOLVE A PROBLEM IN BATCH MODE; AND

EXECUTE-MODE, IN WHICH CYTOS COMMANDS MAY BE READ FROM A SOURCE DATA SET ON A DIRECT ACCESS DISK VOLUME. EXECUTE-MODE CAN BE CONSIDERED AS A CYTOS TASK, BECAUSE IT IS INITIATED BY ISSUING THE CYTOS EXECUTE COMMAND AND TERMINATED BY REVERTING EITHER TO CONVERSATIONAL MODE OR TO CARD-INPUT MODE AFTER THE LAST COMMAND IN THE DATA SET HAS BEEN ACTED UPON.

CARD-INPUT MODE AND EXECUTE-MODE ARE SIMILAR IN MANY RESPECTS. THE MOST SIGNIFICANT OF WHICH IS THAT A PREDETERMINED SET OF CYTOS TASKS IS EXECUTED AS A UNIT. THE FOLLOWING DISCUSSION WILL FAMILIARIZE YOU WITH THE TECHNIQUES USED TO DEFINE SUCH UNITS OF BATCH-ORIENTED PROCESSING.

B. SETTING UP A CARD-INPUT JOB

WHEN YOU SUBMIT A CYTOS JOB IN CARD-INPUT MODE, YOU MUST PREPARE SUFFICIENT CONTROL INFORMATION FOR THE DISK MONITOR SYSTEM TO ACTIVATE CYTOS IN CARD-INPUT MODE. THE CYTOS JOB ITSELF, DEFINED BY THE OCCURRENCE OF THE '// CYTOS' CONTROL RECORD, MAY FOLLOW ANY OTHER VALID DISK MONITOR SYSTEM TASK, SUCH AS A COMPILATION. THE CYTOS JOB, HOWEVER, MUST BE THE LAST TASK OF THE MONITOR JOB, BECAUSE WHEN YOU GIVE THE LOGOFF COMMAND, THE DISK MONITOR SYSTEM WILL FLUSH ALL SUBSEQUENT TASKS UNTIL IT FINDS ANOTHER '// JOB' CONTROL RECORD IN THE BATCH INPUT STREAM.

USED CONVERSATIONALLY, CYTOS IS INVOKED AS DESCRIBED IN SECTION 2.A, "INVOKING THE CYTOS SYSTEM." CYTOS CARD-INPUT IS INVOKED BY PLACING THE OPTION "CARDS" AFTER THE WORD CYTOS ON THE '// CYTOS' CONTROL RECORD. A DISK MONITOR JOB HAVING CYTOS CARD-INPUT AS A TASK MIGHT TAKE THE FOLLOWING FORM:

```
// JOB  (OPTIONAL JOB CARD INFORMATION)
// FOR
     ...
     ... (FORTRAN SOURCE PROGRAM TO COMPILE)
     ...
// XEQ  (TO RUN THE COMPILED PROGRAM)
// CYTOS CARDS
LOGON JONES PROJ2
##       (THE USER'S PASSWORD)
CATPRINT
     ...
     ... (ANY OTHER VALID CYTOS COMMANDS)
     ...
LOGOFF
.END  (END OF CYTOS COMMANDS)
//    (END OF CYTOS COMMAND INPUT)
     ...
     ... (ALL INPUT DECKS TO THE CYTOS JOB, E.G., DECKS
     ... (TO BE READ, COMPILED FROM CARDS, ETC).
//       (END OF CYTOS JOB)
// JOB  (START OF FOLLOWING JOB)
```

IF THE CYTOS JOB IS THE ONLY DMS TASK IN THE DMS JOB, THE '// CYTOS CARDS' RECORD WOULD IMMEDIATELY FOLLOW THE JOB CARD.

IN CARD-INPUT MODE, THE CYTOS SYSTEM INITIALLY READS ALL THE CYTOS COMMANDS INTO A SPECIAL DATA SET CALLED CYTXF. COMMANDS MUST BE TERMINATED WITH A '//' CONTROL RECORD. IF THE DATA SET DOES NOT ALREADY EXIST, SPACE FOR IT WILL BE ALLOCATED AUTOMATICALLY, THE SIZE BEING DEFINED BY THE NUMBER OF COMMANDS GIVEN. IF THE DATA SET ALREADY EXISTS, THE CYTOS JOB WILL BE PROCESSED ONLY IF ALL THE COMMANDS GIVEN CAN BE FIT INTO CYTXF; OTHERWISE, THE JOB WILL FAIL AND YOU WILL HAVE TO DELETE THE CYTXF DATA SET AND INITIATE THE CYTOS JOB AGAIN.

ONCE THE COMMANDS HAVE BEEN PLACED ON DISK, CYTOS WILL READ AND INTERPRET THEM ONE AT A TIME. THE PROMPTING CHARACTERS AND EACH ASSOCIATED COMMAND RESPONSE WILL BE PRINTED ON THE CONSOLE PRINTER TO THAT YOU WILL HAVE A RECORD OF THE SESSION JUST AS IT PROCEEDED. ANY ERROR MESSAGES GENERATED WILL ALSO BE PRINTED. PROCESSING WILL CONTINUE UNTIL EITHER THE LOGOFF COMMAND IS EXECUTED OR A FATAL ERROR IS ENCOUNTERED. IN THE LATTER CASE, SINCE USER-INTERVENTION IS NOT POSSIBLE, CYTOS WILL AUTOMATICALLY LOGOFF AND FLUSH THE REMAINDER OF THE CYTOS JOB LOOKING FOR ANOTHER '// JOB' CONTROL RECORD IN THE INPUT STREAM.

C. SPECIAL CONSIDERATIONS

BECAUSE USER INTERACTION IS OBVIATED BY THE NATURE OF CARD-INPUT CYTOS, SEVERAL FACTORS MUST BE KEPT IN MIND WHEN DEFINING A CARD-INPUT JOB.

1. THE FIRST TWO CARDS OF THE CYTOS TASK SHOULD BE THE LOGON COMMAND AND THE PASSWORD. THE PASSWORD MUST BE LEFT-JUSTIFIED ON THE SECOND CARD. CYTOS WILL NOT ACT UPON FURTHER COMMANDS UNTIL YOU HAVE SPECIFIED VALID LOGON INFORMATION.

2. IN GENERAL, COMMANDS MAY BE PUNCHED IN FREE FORMAT ON THE INPUT CARDS. COMMENTS MAY FOLLOW THE COMMAND, BUT ONLY IF ALL PARAMETERS TO THE COMMAND ARE ENTERED, E.G., CATPRINT HERE () ***COMMENT***. COMMENTS MAY NOT INCLUDE SYNTACTICAL ERRORS, SUCH AS UNBALANCED PARENTHESES OR QUOTES.

3. COMMANDS AND PARAMETERS MAY BE CONTINUED ON SUBSEQUENT RECORDS BY USING THE CONTINUATION CHARACTER '&' IN THE NORMAL WAY.

4. CARD-INPUT COMMAND RECORDS MAY HAVE SEQUENCE NUMBERS IN COLUMNS 73-80. THESE COLUMNS ARE NOT CONSIDERED TO BE PART OF THE INPUT RECORD.

5. UNDER CARD-INPUT AND EXECUTE MODES, CYTOS WILL NOT ALLOW KEYBOARD INPUT TO THE VARIOUS IBM PROCESSORS, SUCH AS FORTRAN. THE .T INPUT DATA SET NAME WILL BE FLAGGED AS AN INVALID OPTION. SIMILARLY, THE READ DSNAME HERE COMMAND IS INVALID. ALL INPUT TO THE SYSTEM MUST BE EITHER FROM CARDS VIA THE PRINCIPAL INPUT/OUTPUT DEVICE (.C) OR FROM THE DISK. '//' (END OF FILE) RECORDS MUST FOLLOW ALL SOURCE CARD FILES TO BE READ, BUT SHOULD NOT FOLLOW ANY OTHER INPUT DECKS SUCH AS CARD SOURCE PROGRAMS TO BE COMPILED.

6. BATCH MODE DMS MAY NOT BE RE-ENTERED VIA THE BATCH COMMAND BECAUSE CYTOS SESSION INTEGRITY CANNOT BE BROKEN.

7. VOLUME MOUNTING IS CONTROLLED NOT BY THE CARTRIDGES INDICATED ON THE '// JOB' CARD, BUT BY SUITABLE PARAMETERS TO THE LOGON AND MOUNT COMMANDS. REMEMBER

THAT THE CYTOS JOB WILL BE LOGGED OFF AUTOMATICALLY IF THE VOLUME ON WHICH THE CYTXF DATA SET IN WHICH THE CYTOS COMMANDS ARE SAVED CAN NO LONGER BE ACCESSED BY CYTOS FOR HAVING BEEN INADVERTENTLY DIS-MOUNT-ED.

8. AS PREVIOUSLY MENTIONED, FATAL ERRORS WILL TERMINATE THE CYTOS JOB AFTER THE APPROPRIATE ERROR MESSAGE HAS BEEN PRINTED ON THE CONSOLE PRINTER. SOME MESSAGES, SUCH AS UT01 NOT FOUND, ARE NOT CONSIDERED FATAL BUT ONLY INFORMATIONAL. IN SUCH CASES, PROCESSING WILL CONTINUE WITH THE NEXT CYTOS COMMAND IN THE CYTXF DATA SET.

9. SPECIAL CARD-INPUT AND EXECUTE MODE PSEUDO-COMMANDS SUCH AS .RESET (DESCRIBED BELOW) MUST BE LEST-JUSTIFIED IN THE COMMAND INPUT RECORD, AS ONLY THE FIRST TWO COLUMNS ARE EXAMINED.

10. PRESSING THE "INT REQ" KEY DURING THE EXECUTION OF A CARD-INPUT COMMAND HAS THE SAME EFFECT AS IN CONVERSATIONAL MODE; CONTROL IS RETURNED TO THE COMMAND LANGUAGE INTERPRETER TO ACCEPT THE NEXT CYTOS COMMAND IN CONVERSATIONAL MODE. THE CYTOS JOB IS NOT FLUSHED AS IS A NORMAL DMS BATCH JOB UNDER SIMILAR CIRCUMSTANCES.

11. INPUT TO THE EDITOR'S INPUT-MODE MUST BE EMBEDDED WITHIN THE COMMAND FILE ITSELF (RATHER THAN AFTER THE '//' DELIMITER); ONLY INPUT THAT IS NOT EXPECTED AS THE RESPONSE TO A CYTOS PROMPT MAY FOLLOW THE COMMAND FILE '//' DELIMITER.

D. SPECIAL CARD-INPUT PSEUDO-COMMANDS

IT IS NECESSARY IN SOME CASES TO SIMULATE USER INTERVENTION IN CYTOS CARD-INPUT PROCESSING OR EXECUTE-MODE PROCESSING EVEN THOUGH THE NATURE OF THE JOB IS BATCH-ORIENTED. THE CYTOS PSEUDO-COMMANDS LISTED BELOW MAY BE USEFUL FOR SUCH APPLICATIONS. ALL PSEUDO-COMMANDS MUST BE LEFT-JUSTIFIED IN THE COMMAND INPUT RECORD.

.ATTENTION (OR .A)

THIS COMMAND MUST BE USED WHEN CYTOS IS LOGICALLY AT COMMAND LEVEL AND EXPECTS AN ATTENTION (I.E., PRESSING THE "INT REQ" KEY) AS A VALID RESPONSE, AS WHEN YOU WISH TO LEAVE INSERT-MODE FOR EDIT-MODE AFTER INSERTING ADDITIONAL INFORMATION IN A DATA SET WITH THE CYTOS EDITOR. THIS COMMAND CANNOT BE RECOGNIZED BY USER COMMANDS WHICH ACCEPT INFORMATION FROM THE KEYBOARD, HOWEVER, BECAUSE SUCH INPUT IS NOT CONSIDERED TO BE ENTERED AT CYTOS COMMAND LEVEL.

.END (OR .E)

THE .END COMMAND MUST BE USED TO SIGNAL THE END OF VALID CYTOS COMMANDS IN A DATA SET. THIS FEATURE IS MOST USEFUL IN EXECUTE-MODE TO SIGNAL A RETURN TO CONVERSATIONAL MODE. IT MAY ALSO BE USED IN LIEU OF A LOGOFF COMMAND FOR CYTOS CARD-INPUT JOBS. IN CARD-INPUT MODE, A LOGOFF IS AUTOMATICALLY GENERATED.

.RESET (OR .R)

THIS PSUEDO-COMMAND MAY BE USED TO RESET THE ERROR CONDITION CAUSED BY AN INVALID CYTOS COMMAND WHICH WOULD NORMALLY REQUIRE THE FLUSHING OF THE CYTOS CARD-INPUT JOB OR EXECUTE FILE COMMAND INPUT. IF ENCOUNTERED IN THE COMMAND STREAM, CYTOS WILL AGAIN START TO ACCEPT COMMANDS, BEGINNING WITH THE VERY NEXT COMMAND AFTER THE .RESET, AS IF NO FATAL ERROR HAD BEEN ENCOUNTERED. THIS FACILITY IS USEFUL, FOR INSTANCE, TO CONTINUE A CYTOS JOB EVEN THOUGH A PARTICULAR COMPILATION PRODUCED ERRORS THAT WOULD NORMALLY PRECLUDE FURTHER PROCESSING.

THE FOLLOWING EXAMPLE SHOWS HOW EACH OF THE ABOVE RECORDS MIGHT BE USED IN A CARD-INPUT CYTOS JOB:

```
// JOB
// CYTOS CARDS
LOGON JONES PROJ2
##
EDIT NEWFR FORTRAN
C
C  FORTRAN PROGRAM WITH AN ERROR
C
      WRITE (1,10)
20    FORMAT('INCORRECT FORMAT STATEMENT NUMBER')
      CALL EXIT
      END
.ATTENTION  (RETURNS TO EDIT-MODE)
TOP, PRINT 10
FILE
FORTRAN NEWPR .C (LIST ALL HERE)  'CAUSES CYTOS UT06 ERROR'
PRINT NEWPR HERE ()       (COMMAND WILL BE SKIPPED)
CATPRINT () ()            (COMMAND WILL BE SKIPPED)
```

```
.RESET                          (RESETS ERROR CONDITION)
FLUSH                           (BYPASS BLANK CARDS)
ERASE NEWPR.S                   (COMMAND EXECUTED REGARDLESS OF ERRORS)
LOGOFF
.END                            (SIGNALS END OF INPUT COMMANDS)
//                              (ANY // CARD SIGNALS END OF FILE)
  ...
  ...       (BLANK CARDS PUNCHED IF COMPILATION SUCESSFUL)
  ...
//                              (END OF CYTOS JOB)
// JOB                          (... NEXT JOB)
```

E. A WORD ABOUT 'EXECUTE-MODE'

A DATA SET CONTAINING COMMANDS TO BE INTERPRETED UNDER EXECUTE MODE (BY ISSUING THE EXECUTE DSNAME COMMAND) IS FORMED MUCH IN THE SAME WAY AS A CYTOS CARD-INPUT JOB. THE DIFFERENCES BETWEEN THE TWO MODES ARE ENUMERATED HERE:

1. THE DATA SET CONTAINING CYTOS COMMANDS TO BE EXECUTED SHOULD NOT INCLUDE ANY DMS MONITOR CONTROL CARDS, AS THESE ARE NOT VALID CYTOS COMMANDS.

2. YOU SHOULD NOT INCLUDE THE LOGON COMMAND OR ASSOCIATED PASSWORD RECORD, AS YOU MUST BE LOGGED ON ALREADY EVEN TO ISSUE THE EXECUTE COMMAND.

3. YOU MAY INCLUDE A LOGOFF COMMAND AT THE END OF THE EXECUTE FILE IF YOU WISH TO RETURN CONTROL TO THE DISK MONITOR SYSTEM IN BATCH MODE (CYTOS/I), OR TO DISCONNECT THE TERMINAL (CYTOS/II); OTHERWISE, YOUR LAST COMMAND MUST BE KEYBOARD, OR .END, IN ORDER TO RETURN CONTROL TO CYTOS IN 'CONVERSATIONAL' MODE.

4. IF YOU DO NOT HAVE EITHER A LOGOFF OR KEYBOARD COMMAND, OR PSEUDO-COMMAND .END, A LOGOFF WILL BE GENERATED AUTOMATICALLY.

5. ALL 80 COLUMNS OF THE EXECUTE FILE RECORD ARE SCANNED BY THE CYTOS COMMAND LANGUAGE INTERPRETER. IF THE FILE CONTAINS SEQUENCE NUMBERS, YOU MAY LIMIT THE SCANNING BY SPECIFYING THE LAST SCAN COLUMN, AS FOR EXAMPLE:

```
==> EXECUTE NCOMP PRINTER 72
```

9. CYTOS CONVENTIONS AND COMMANDS

A. DATA SET NAMING CONVENTIONS

A DATA SET NAME UNIQUELY IDENTIFIES A DATA SET TO THE CYTOS SYSTEM. 'SIMPLE DATA SET NAMES' ARE ONE TO EIGHT CHARACTERS. THE FIRST OF WHICH MUST BE ALPHABETIC. ONLY THE FIRST FIVE CHARACTERS OF THE NAME ARE SIGNIFICANT TO THE CYTOS/1130 SYSTEM. BUT EIGHT CHARACTERS ARE ALLOWED FOR THE SAKE OF COMPATABILITY WITH CYTOS ON OTHER SYSTEMS. SECOND AND SUCCEEDING CHARACTERS MAY BE ANY OF THE FOLLOWING:

THE LETTERS "A" TO "Z"
THE DIGITS "0" TO "9"
THE SPECIAL CHARACTERS "$" AND "#"

FOR EXAMPLE, XX#14 IS A VALID SIMPLE DATA SET NAME.

TO INDICATE WHAT TYPE OF INFORMATION IS TO BE FOUND IN THE DATA SET, THE SIMPLE DATA SET NAME MAY BE FOLLOWED BY A PERIOD AND ONE OF THE LETTERS S, O, OR L:

XX#14.S CONTAINS A SOURCE PROGRAM, OR DATA INPUT TO OR OUTPUT FROM A CYTOS COMMAND OR USER COMMAND.

CURVE.O CONTAINS AN OBJECT MODULE, THE OUTPUT OF AN IBM LANGUAGE PROCESSOR SUITABLE FOR LINKING TO OTHER OBJECT MODULES TO FORM AN EXECUTABLE PROGRAM.

CURVE.L CONTAINS A LOAD MODULE, THE OUTPUT OF THE IBM CORE LOAD BUILDER. LOAD MODULES ARE IMMEDIATELY EXECUTABLE PROGRAMS.

THUS THERE CAN BE THREE DATA SETS WITH THE SAME SIMPLE DATA SET NAME; THAT IS, ONE FOR EACH TYPE. IT IS NOT ALWAYS NECESSARY TO SPECIFY THE TYPE WHEN REFERRING TO ONE OF YOUR OWN DATA SETS. BECAUSE THE SYSTEM WILL ASSUME WHAT TYPE OF DATA SET IS BEING NAMED WHENEVER POSSIBLE. FOR CERTAIN SYSTEM COMMANDS (E.G., COPY AND ERASE) ANY TYPE IS ACCEPTABLE; IF NO TYPE IS SPECIFIED, .S IS ASSUMED. (ERASE REQUIRES THE TYPE TO BE SPECIFIED.)

.O IS ASSUMED WHEN A SIMPLE DATA SET NAME IS GIVEN AS OUTPUT OF ANY OF THE COMPILERS, INPUT TO THE CORE LOAD BUILDER (LINKEDIT AND LOADGO COMMANDS, AND PROGRAMS REFERENCED BY THE CORE LOAD SPECIFICATION COMMANDS INCLUDE, OVERLAY, SUBSTITUTE, AND GRAPHICS);

.L IS ASSUMED WHEN A SIMPLE DATA SET NAME IS GIVEN AS OUTPUT OF THE CORE LOAD BUILDER (LINKEDIT), OR AS INPUT TO THE RUN COMMAND.

.S IS ASSUMED IN ALL OTHER CASES WHERE AN ASSUMPTION IS POSSIBLE.

SOME COMMANDS (E.G., COPY, DESCRIBE) ACCEPT DATA SET NAMES OF THE FORM 'DSNAME.*'. THE ASTERISK INDICATES 'ALL' TYPES OF THE DATA SET. FOR EXAMPLE:

CZ$2.* MEANS CZ$2.S, CZ$2.O, AND CZ$2.L.

TO USE DATA SETS CREATED AND OWNED BY OTHER USERS, YOU MUST USE THE 'FULLY QUALIFIED DATA SET NAME'

USERID.DSNAME.TYPE

WHERE DSNAME IS THE SIMPLE DATA SET NAME, TYPE IS THE DATA SET TYPE, AND USERID IS THE OWNER OF THE VOLUME ON WHICH THE DATA SET IS CATALOGUED. IF A USER NAMED SMITH WANTS TO COMPILE A FORTRAN SOURCE PROGRAM NAMED REGRS CREATED BY JONES, HE WOULD TYPE THE FOLLOWING:

==> FORTRAN JONES.REGRS.S MYOBJ (ALL HERE)

BEFORE ONE PROGRAMMER MAY USE THE DATA SETS OF ANOTHER, PERMISSION TO DO SO MUST PREVIOUSLY HAVE BEEN GIVEN BY THE OWNER (SEE THE 'PERMISSION TUTORIAL,' SECTION 6 ABOVE, AND THE PERMIT AND UNPERMIT COMMANDS). WHEN A USER IS REFERRING TO ONE OF HIS OWN DATA SETS HE NEED NOT QUALIFY IT WITH HIS OWN USERID: WHEN REFERRING TO SOMEONE ELSE'S DATA SETS, HOWEVER, THE SIMPLE DATA SET NAME MUST BE QUALIFIED WITH BOTH THE USERID AND THE DATA SET TYPE.

IF YOU ARE LOGGED ONTO A SYSTEM WHICH HAS ACCESS TO MULTIPLE DIRECT ACCESS VOLUMES, AND MORE THAN ONE OF YOUR OWN VOLUMES IS AVAILABLE TO YOU, YOU MUST FURTHER QUALIFY THE SIMPLE DATA SET NAME OF ALL DATA SETS ON ALL VOLUMES EXCEPT THE VOLUME WITH WHICH YOU ORIGINALLY LOGGED ON (I.E., THE VOLUME FROM WHICH A VALID COMPARISON WITH YOUR USERID, PROJID, AND PASSWORD WAS MADE). TO INDICATE THAT A DATA SET IS ON A PARTICULAR NON-LOGON VOLUME, YOU MAY SPECIFY EITHER THE CARTRIDGE_ID OF THE VOLUME OR THE LOGICAL DRIVE NUMBER OF THE UNIT ON WHICH THE VOLUME WAS MOUNTED:

MYPRG.S(2222) -- DATA SET ON VOLUME WITH ID 2222
MYPRG.S(1) -- DATA SET ON LOGICAL DRIVE 1.

SIMILARLY, YOU NEED QUALIFY THE DATA SET NAME OF ANOTHER USER'S DATA SET WITH THE SPECIFIC CARTRIDGE_ID OR LOGICAL DRIVE NUMBER ONLY IF THAT VOLUME IS NOT THE FIRST (IN LOGICAL DRIVE ORDER) OF ALL VOLUMES CURRENTLY MOUNTED THAT BELONG TO THAT PARTICULAR USER. THE LOGICAL ORDER OF VOLUMES ON DRIVES IS DEFINED BY THE LIST OF CARTRIDGE_IDS SPECIFIED WITH EITHER THE LOGON OR MOUNT COMMANDS. FOR EXAMPLE:

ASSUME THAT USER SMITH OWNS VOLUMES 2222 AND 3333, AND THAT USER JONES OWNS VOLUMES 1001, 1002, AND 1003. FURTHER, ASSUME THAT JONES IS CURRENTLY LOGGED ON AND HAS ISSUED THE CYTOS COMMAND:

```
==> MOUNT (1001 2222 1002 1003 3333)
```

THEN, ASSUMING THAT DATA SETS OF THE SAME NAME EXIST ON ALL FIVE VOLUMES, JONES WOULD HAVE TO USE THE FOLLOWING MINIMUM DATA SET NAMES:

```
FORTA IF REFERRING TO VOLUME 1001, HIS OWN.
     (OR: FORTA.S, FORTA.S(0), FORTA(1001),
     FORTA.S(1001), JONES.FORTA.S,
     JONES.FORTA.S(0), OR
     JONES.FORTA.S(1001)).

SMITH.FORTA.S IF ON VOLUME 2222
     (OR: FORTA(2), FORTA.S(1002),
     FORTA.S(2), JONES.FORTA.S(2),
     JONES.FORTA.S(1002))

FORTA(1003) IF REFERRING TO VOLUME 1003
     (OR: FORTA(3), FORTA.S(1003),
     FORTA.S(3), JONES.FORTA.S(3),
     JONES.FORTA.S(1003))

SMITH.FORTA.S(3333) IF ON VOLUME 3333
     (OR: SMITH.FORTA.S(4))
```

IN OTHER WORDS, ONLY ENOUGH INFORMATION TO UNIQUELY IDENTIFY THE DATA SET NEED BE GIVEN. IN MOST CASES, THE SIMPLE DATA SET NAME WITHOUT ANY QUALIFICATIONS WILL SUFFICE. IN GENERAL, IF NO VOLUME INFORMATION IS GIVEN, THE FIRST DATA SET OF THE SPECIFIED NAME AND TYPE FOUND ON ANY OF THE VOLUMES BELONGING TO THE USER (SPECIFIED OR IMPLIED), AS THOSE VOLUMES ARE SEARCHED IN LOGICAL-DRIVE-ORDER, WILL BE USED.

IN CERTAIN CASES IT MAY BE NECESSARY TO REFER TO A DATA SET THAT CAN ONLY EXIST IN A "FIXED" LIBRARY (THE SO-CALLED "FIXED AREA" UNDER THE DISK MONITOR SYSTEM). SEARCHING FOR A DATA SET NAME WILL BE LIMITED TO THE FIXED LIBRARY ONLY IF YOU QUALIFY THE DATA SET NAME WITH THE CARTRIDGE_ID PRECEDED WITH A COLON, AS IN THE FOLLOWING EXAMPLE.

```
FDATA.S(:2222)
```

THE FIXED SPECIFICATION NEED BE GIVEN ONLY FOR COMMANDS THAT CREATE NEW DATA SETS, TO INDICATE THAT THEY MUST BE PLACED IN THE FIXED LIBRARY. IF REFERRING TO AN OLD DATA SET, BOTH THE FIXED LIBRARY AND THE USER'S LIBRARY ("USER AREA") ARE SEARCHED FOR THE SPECIFIED DATA SET (AS WITH THE PRINT COMMAND). TWO DATA SETS OF

B. FORMAT OF COMMAND DESCRIPTIONS

IN THE FOLLOWING SECTION, ALL VALID CYTOS SYSTEM COMMANDS ARE LISTED WITH THEIR VALID PARAMETERS AND A SHORT EXPLANATION OF THE FUNCTION OF EACH. EXAMPLES ARE GIVEN FOR TYPICAL CASES IN WHICH THE COMMAND MIGHT BE USED. THE NOTATION FOR PRESENTING THESE COMMANDS IS AS FOLLOWS:

COMMAND VERBS, MUST BE ENTERED EXACTLY AS GIVEN:	COMMAND
PARAMETERS WHICH NEED NOT BE GIVEN, I.E. ARE OPTIONAL, ARE SURROUNDED BY BRACKETS (THE BRACKETS ARE NOT ENTERED AS PART OF THE TYPED COMMAND);	COMMAND <OPTIONAL_PARM>
IF A PARAMETER MAY BE A COLLECTION, THIS WILL BE NOTED BY PARENTHESES. THE PARENTHESES NEED BE ENTERED ONLY IF THERE IS MORE THAN ONE ITEM IN A COLLECTION.	(DSNAME,...)
IF A CHOICE MUST BE MADE BETWEEN TWO OR MORE KEYWORD OPTIONS, THE POSSIBILITIES WILL BE SEPARATED BY VERTICAL BARS.	OPTA|OPTB|OPTC
A DEFAULT CONDITION WILL BE UNDERLINED.	OPTA|OPTB ----

THE SAME NAME AND TYPE CANNOT THEREFORE EXIST ONE IN THE FIXED LIBRARY AND ONE IN THE USER'S LIBRARY. THIS IS NOT TO SAY THAT A SOURCE DATA SET CANNOT BE PLACED IN THE FIXED LIBRARY AND AN OBJECT MODULE OF THE SAME NAME IN THE USER'S LIBRARY, AS FOR INSTANCE BY THE FOLLOWING COMMAND:

```
==> RESTORE (ACREC.S(:2222),ACREC.O) (240,35)
```

A TABLE IN APPENDIX B LISTS ALL DATA SETS BY COMMAND AND INDICATES HOW EACH MAY BE QUALIFIED AS REGARDS TYPE, OWNERSHIP, AND LIBRARY LOCATION.

SEVERAL DATA SET NAMES ARE RESERVED FOR SPECIAL SITUATIONS:

".T" (OR ".TERMINAL") IS USED WHENEVER INPUT TO AN IBM LANGUAGE PROCESSOR IS TO ORIGINATE FROM THE TERMINAL.

".C" (OR ".CARDS") MAY BE SPECIFIED WHENEVER INPUT TO AN IBM LANGUAGE PROCESSOR IS TO ORIGINATE FROM CARDS, OR WHEN AN OBJECT DECK IS TO BE PUNCHED AS THE OUTPUT OF THE COMPILATION.

".D" (OR ".DUMMY") MAY BE GIVEN IF THE OUTPUT FROM A COMPILER IS NOT TO BE SAVED IN ANY FORM.

"#" MAY BE GIVEN WHENEVER A DATA SET IS TO BE USED ONLY DURING THE EXECUTION OF A USER-WRITTEN PROGRAM AND THEN IS TO BE DISCARDED (A SCRATCH DATA SET). THIS NAME IS ONLY APPLICABLE TO THE NAMEFILE COMMAND AND RESPONSE TO THE WHATFILE QUERY.

SUMMARY: THE VARIOUS COMPONENTS OF A FULLY QUALIFIED DATA SET NAME, WHEN ALL CONCATENATED INTO ONE PATTERN, FORM THE FOLLOWING GENERAL TYPE:

```
USERID.DSNAME.TYPE(:VVVV)
```

WHERE:

USERID IS THE OWNER'S NAME
DSNAME IS THE SIMPLE DATA SET NAME
TYPE REPRESENTS THE DATA SET FORM (S, O, L, OR *)
(:VVVV) REPRESENTS THE LOCATION --
VVVV TO SPECIFY A CARTRIDGE ID,
N TO SPECIFY A LOGICAL DRIVE NUMBER:
: RESTRICTS DATA SET TO FIXED LIBRARY.

C. CYTOS SYSTEM COMMAND DESCRIPTIONS

```
ALLOCATE (DSNAME, ...) <ALLOC_UNIT> <SIZE>
```

FUNCTION: RESERVE DISK STORAGE SPACE FOR ONE OR MORE SOURCE DATA SETS.

ALIAS: DFILE

PARAMETERS:

DSNAMES — IF A COLLECTION OF NAMES IS GIVEN, IDENTICAL DATA SETS WILL BE CREATED.

ALLOC_UNIT — THE SIZE OF THE NEW DATA SETS IS SPECIFIED IN TERMS OF ONE OF THE FOLLOWING UNITS:

BLOCKS|CARDS|SECTORS|TRACKS|CYLINDERS

SIZE — THE NUMBER OF UNITS TO BE ALLOCATED FOR EACH DATA SET. IF OMITTED, 80 UNITS OF THE SPECIFIED TYPE WILL BE ALLOCATED.

EXAMPLE: ==> ALLOCATE (CRV1,CRV2,CRV3)

THREE SOURCE DATA SETS, NAMED CRV1, CRV2, AND CRV3 WILL BE CREATED. SINCE NO SPECIFIC SIZE IS GIVEN, THE DEFAULT SIZE (40 CARDS) WILL BE USED FOR EACH.

NOTES: (1) IF A DATA SET IS MODIFIED WITH ANOTHER USER'S ID, PERMISSION TO WRITE ON HIS VOLUME MUST PREVIOUSLY HAVE BEEN GRANTED FOR THE DATA SET TO BE CREATED.

(2) THE MAXIMUM SIZE OF A FILE TO BE CREATED IS THE SIZE OF WORKING STORAGE ON THE OBJECT VOLUME AT THE TIME THE ALLOCATE COMMAND IS ISSUED. THIS DIFFERS SIGNIFICANTLY FROM THE NEWSPACE COMMAND, WHICH ACTUALLY CAUSES THE CONTENTS OF SYSTEM WORKING STORAGE TO BE MOVED TO THE WORKING STORAGE OF THE OBJECT VOLUME BEFORE THE CATALOGUE (LET) ENTRY IS MADE.

(3) IF THE ROUND OPTION HAS PREVIOUSLY BEEN SET, THE SOURCE FILE CREATED WILL BE PADDED WITH SUFFICIENT RECORDS TO FORCE THE SIZE OF THE FILE TO AN INTEGRAL SECTOR BOUNDARY.

```
ASSEMBLE  INDSNAME  <OUTDSNAME>
          <OPTIONS>  <MACLIB_NAME>
```

FUNCTION: INVOKE THE IBM MACRO ASSEMBLER TO PRODUCE AN OBJECT MODULE FROM AN ASSEMBLER SOURCE PROGRAM.

ALIASES: ASM, A

PARAMETERS:

INDSNAME — INDSNAME NAMES THE SOURCE DATA SET CONTAINING THE ASSEMBLER SOURCE PROGRAM. ONLY ONE DATA SET IS ALLOWED. ".T" MAY BE GIVEN IF YOU INTEND TO ENTER THE SOURCE PROGRAM FROM THE KEYBOARD, OR ".C" IF THE SOURCE PROGRAM IS TO BE READ FROM CARDS.

OUTDSNAME — OUTDSNAME IS THE NAME OF THE OBJECT MODULE PRODUCED. THE TYPE NEED NOT BE GIVEN; .O IS ASSUMED. IF OUTDSNAME IS OMITTED, NO OBJECT MODULE WILL BE PRODUCED; IF A NULL PARAMETER "()" IS SPECIFIED, AN OBJECT MODULE NAMED "INDSNAME.O" WILL BE CATALOGUED. IF OUTDSNAME.O ALREADY EXISTS, IT WILL BE REUSED, SO LONG AS THE NEW VERSION IS NO LARGER THAN THE OLD. IF YOU WISH TO SAVE THE OBJECT MODULE ON CARDS, YOU MAY SPECIFY ".C"; IF YOU WANT NO OBJECT MODULE AT ALL, YOU MAY SPECIFY ".D" (OR ".DUMMY").

OPTIONS — THE FOLLOWING OPTIONS ARE ALLOWED, AND WILL HAVE THE INDICATED EFFECTS:

LIST — LIST THE SOURCE PROGRAM.

HERE — ALL PRINTED OUTPUT WILL GO TO THE TERMINAL INSTEAD OF TO THE PRINCIPAL PRINTER.

XREF — PRODUCE A CROSS-REFERENCE TABLE AS PART OF THE PRINTED OUTPUT.

SYMBOLS — PRINT THE SYMBOL TABLE.

LEVEL(N) — INDICATES THAT THE PROGRAM IS AN INTERRUPT SERVICE SUBROUTINE DESIGNED TO HANDLE INTERRUPTS ON THE INDICATED LEVEL (0 THROUGH 5).

OVERFLOW(N) — PROVIDE N DISK SECTORS FOR SYMBOL TABLE OVERFLOW.

MACROS(N) — PROVIDE N SECTORS FOR TEMPORARY MACRO DEFINITIONS.

PARMS(N) PROVIDE N SECTORS FOR MACRO PARAMETER SPILL.

COMMON(N) INDICATE THAT THE ASSEMBLED PROGRAM REQUIRES A COMMON AREA OF AT LEAST N WORDS DURING EXECUTION.

TWOPASS TOW PASS MODE -- THE SOURCE PROGRAM IS READ TWICE SO AS TO CONSERVE SPACE IN SYSTEM WOKRING STORAGE (VALID FOR CARD OR DISK INPUT ONLY).

USE INCLUDE A PREVIOUSLY DEFINED "SYSTEM SYMBOL TABLE".

SAVE SAVE THE CURRENT SET OF SYMBOLS IN THE SYSTEM SYMBOL TABLE SO THAT THEY MAY BE USED BY FUTURE ASSEMBLIES.

MACLIB_NAME THE NAME OF A VALID MACRO LIBRARY TO BE USED FOR THIS ASSEMBLY. IF YOU WISH TO USE MORE THAN ONE LIBRARY, SUCCESSIVE LIBRARIES MUST HAVE PREVIOUSLY BEEN CONCATENATED TO THE SPECIFIED LIBRARY, USING THE MACRO UPDATE FACILITY VIA THE MACLIB COMMAND.

REFERENCE: FOR FURTHER INFORMATION ABOUT SPECIFIC OPTIONS FOR THE ASSEMBLER, SEE "IBM 1130/1800 ASSEMBLER LANGUAGE", GC26-3778.

MESSAGES:
NNN OVERFLOW SECTORS SPECIFIED
NNN OVERFLOW SECTORS REQUIRED
NNN SYMBOLS DEFINED
NNN ERROR(S) AND NNN WARNING(S) FLAGGED IN ABOVE ASSEMBLY

EXAMPLE: ==> A AVRGE .C (LIST,XREF,MACROS(3)) DAVIS.MACRO.S

THE SOURCE PROGRAM AVRGE IS TO BE ASSEMBLED, AND IF SUCCESSFUL, THE OBJECT MODULE IS TO BE PUNCHED TO CARDS. THE ASSEMBLER WILL USE DAVIS.MACRO.S AS ITS PRIMARY MACRO LIBRARY, BUT WILL ALSO RESERVE 3 SECTORS FOR TEMPORARY MACRO DEFINITIONS WITHIN THE SOURCE PROGRAM. A LIST AND CROSS-REFERENCE TABLE WILL BE INCLUDED IN THE PRINTED OUTPUT.

NOTES: (1) THE MACRO FEATURES OF THE MACRO ASSEMBLER MAY NOT BE USED ON 4K SYSTEMS.

(2) THE FOLLOWING OPTIONS ARE NOT SUPPORTED FOR INPUT FROM DISK:

LIST DECK
LIST DECK E
PUNCH SYMBOL TABLE

(3) ANY OPTIONS WHICH ARE VALID MAY BE INCLUDED AS ASSEMBLER CONTROL CARDS AT THE BEGINNING OF THE SOURCE PROGRAM, IN THE USUAL MANNER, THOSE THAT ARE SPECIFIED AT THE TIME THE ASSEMBLE COMMAND IS

ISSUED WILL BE IN ADDITION TO THOSE IN THE SOURCE PROGRAM. TO DELETE OPTIONS FROM THE SOURCE PROGRAM, YOU MUST USE THE EDITOR.

(4) IT IS HIGHLY RECOMMENDED TO SUPPRESS SOURCE PROGRAM LISTING AS MUCH AS POSSIBLE WHEN ASSEMBLING FROM DISK IN TWOPASS MODE; THE SYSTEM PRINTER SUBROUTINE MUST BE RELOADED BY THE ASSEMBLER EVERY TIME A LINE IS TO BE PRINTED.

ASSIGN OPTION CARTRIDGE_ID

FUNCTION: ASSIGN A SYSTEM WORK AREA TO A SPECIFIC VOLUME.

PARAMETERS:

OPTION — ONLY ONE OF THE FOLLOWING MAY BE SPECIFIED FOR EACH ISSUANCE OF THE COMMAND:

WS	SYSTEM WORKING STORAGE
CIB	SYSTEM CORE IMAGE BUFFER
UFIO	FORTRAN UNFORMATTED I/O AREA -- A DATA SET NAMED $$$$$ MUST EXIST ON THE SPECIFIED VOLUME.

CART_ID — SPECIFIES THE ID OF THE VOLUME ON WHICH THE DESIGNATED AREA IS TO BE ASSIGNED. IF THE VOLUME IS NOT MOUNTED AT THE TIME THE ASSIGN COMMAND IS GIVEN, THE AREA WILL BE PLACED BY DEFAULT ON THE MASTER CARTRIDGE (I. E., THE VOLUME CURRENTLY MOUNTED ON LOGICAL DRIVE 0 FOR CYTOS/I OR LOGICAL DRIVE 1 FOR CYTOS/II).

EXAMPLE:

```
==> ASSIGN WS 020A
```

SYSTEM WORKING STORAGE WILL BE ASSIGNED (OR RE-ASSIGNED) TO THE VOLUME WHOSE ID IS 020A, ASSUMING THAT CARTRIDGE IS CURRENTLY ACCESSIBLE TO THE SYSTEM.

NOTE: IF THE SPECIFIED VOLUME IS DISMOUNTED BECAUSE OF A LATER MOUNT COMMAND, THE WORK AREA THUS ASSIGNED WILL BE RE-ASSIGNED AUTOMATICALLY TO THE MASTER CARTRIDGE SPECIFIED IN THE MOUNT COMMAND. YOU MAY THEN ISSUE THE ASSIGN COMMAND AGAIN IF YOU WISH TO PLACE THE AREA ON ANOTHER SATELLITE VOLUME.

BATCH

FUNCTION: ENTER DISK MONITOR SYSTEM BATCH MODE FROM CONVERSATIONAL MODE, AND CAUSE A BATCH JOB STREAM TO BE PROCESSED FROM THE SYSTEM INPUT DEVICE.

ALIAS: B, DMS

NOTE: THE FIRST "// CYTOS" MONITOR CONTROL CARD ENCOUNTERED IN THE JOB STREAM WILL RETURN CONTROL TO CONVERSATIONAL MODE.

CATALOGUE DSNAME <NDUP> <NN>

FUNCTION: MOVE A DATA SET CURRENTLY IN SYSTEM WORKING STORAGE TO A USER LIBRARY, AND MAKE AN ENTRY FOR THE DATA SET IN THE LOCATION EQUIVALENCE TABLE (LET).

PARAMETERS:

DSNAME — THE NAME TO BE GIVEN TO THE DATA SET ONCE IT HAS BEEN MOVED INTO THE USER'S LIBRARY. ANY TYPE OF DATA SET MAY BE CATALOGUED; HOWEVER, THE TYPE MUST AGREE WITH THAT OF THE DATA SET CURRENTLY IN SYSTEM WORKING STORAGE.

NDUP — BY SPECIFYING THIS OPTION, YOU MAY PREVENT CATALOGUING AFTER A COMPILATION OR ASSEMBLY WHICH WAS UNSUCCESSFUL DUE TO ERRORS ENCOUNTERED IN THE SOURCE PROGRAM. BY CATALOGUING THE DATA SET, YOU MAY USE THE OBJECT MODULE IN SPITE OF THESE ERRORS, BUT SHOULD BE AWARE THAT THEY MAY CAUSE ABNORMAL TERMINATION OF YOUR PROGRAM ONCE IT HAS BEEN PLACED IN EXECUTION.

NN — ANY VALID SUBROUTINE SUB-TYPE MAY BE SPECIFIED AS THE THIRD PARAMETER IF YOU WISH THIS OBJECT MODULE TO BE INCLUDED IN THE SPECIAL SYSTEM OVERLAY AREA WHEN IT IS REFERENCED DURING THE BUILDING OF A CORE LOAD.

NOTE: YOU MAY NOT CATALOGUE A DATA SET IF THE SPECIFIED NAME IS ALREADY IN USE FOR ANOTHER DATA SET. SEE THE "RECATALOGUE" COMMAND IF YOU WISH TO REPLACE AN OLD DATA SET WITH A NEWER VERSION.

CATPRINT <HERE> <CART_ID>

FUNCTION: PRINT INFORMATION FROM A USER'S CATALOGUE.

PARAMETERS: IF NO PARAMETERS ARE GIVEN, ALL CATALOGUES ON CURRENTLY MOUNTED USER VOLUMES WILL BE PRINTED, REGARDLESS OF OWNERSHIP.

HERE — INDICATES THAT CATALOGUES ARE TO BE PRINTED AT THE TERMINAL INSTEAD OF ON THE PRINCIPAL PRINTER (THE DEFAULT).

CART_ID — PRINTING MAY BE LIMITED TO ONLY ONE LIBRARY, NAMELY THAT ON THE VOLUME WHOSE CARTRIDGE IDENTIFICATION IS GIVEN AS THE SECOND PARAMETER.

EXAMPLE: ==> CATPRINT () 00A6

PRINT THE CONTENTS OF THE LIBRARY CATALOGUE ON VOLUME 00A6 ON THE PRINCIPAL PRINT DEVICE.

NOTES:

(1) IF YOU HAVE ONLY ONE DISK-DRIVE ON YOUR SYSTEM, YOU NEED NOT SPECIFY THE CARTRIDGE IDENTIFICATION FOR THE VOLUME ON THAT DRIVE, AS PRINTING ALL VOLUMES WILL BE EQUIVALENT TO PRINTING JUST THE ONE.

(2) IF FIXED AREAS HAVE BEEN DEFINED, CATALOGUES FOR EACH FIXED LOCATION EQUIVALENCE TABLE (FLET) WILL BE PRINTED AFTER THE NORMAL LOCATION EQUIVALENCE TABLE (LET) ON ANY VOLUME WHERE THE TWO LIBRARIES CO-EXIST.

(3) THE DATA SET TYPE .X ON THE PRINTED OUTPUT INDICATES AN UNUSABLE PADDING AREA GENERATED IN ORDER TO ALIGN .S AND .L DATA SETS TO A DISK SECTOR BOUNDARY.

CHECKPOINT DSNAME

FUNCTION: SAVE THE ENTIRE STATUS OF A PROGRAM THAT HAS BEEN INTERRUPTED BY PRESSING THE "INT REQ" KEY. THE "RESTART" COMMAND MAY BE USED AT A LATER TIME TO BRING THE PROGRAM BACK INTO CORE IN PREPARATION FOR RESUMPTION.

ALIAS: CP

PARAMETER:

DSNAME THE NAME OF A DATA SET IN WHICH THE CORE IMAGE OF THE INTERRUPTED PROGRAM MAY BE SAVED. THE SIZE OF DSNAME MUST BE AT LEAST AS LARGE AS THE SIZE OF YOUR CORE, PLUS THREE SECTORS:

```
 4K =  16 SECTORS
 8K =  29 SECTORS
16K =  55 SECTORS
32K = 106 SECTORS
```

THE DATA SET MUST EXIST AT THE TIME THE CHECKPOINT COMMAND IS GIVEN.

EXAMPLE: ==> CHECKPOINT SAVEFILE(6666)

THE INTERRUPTED PROGRAM WILL BE SAVED IN THE SOURCE DATA SET SAVEF.S ON CARTRIDGE 6666. THE PROGRAM COULD BE RESUMED LATER BY ISSUING THE FOLLOWING COMMAND:

==> RESTART SAVEFILE.S(6666) RESUME

NOTES: (1) UNDER CERTAIN CONDITIONS AN INTERRUPTED AND CHECKPOINTED PROGRAM CANNOT BE RESUMED. RESUMPTION IS NOT POSSIBLE IF EITHER THE LOCATION OR THE SIZE OF SYSTEM WORKING STORAGE ON ANY MOUNTED VOLUMES HAS CHANGED BETWEEN THE PROGRAM INTERRUPTION AND RESUMPTION. IF THE INTERRUPTED PROGRAM HAD BEEN USING WORKING STORAGE, YOU MUST ISSUE A NEWSPACE OF SUFFICIENT SIZE TO PROTECT ITS CONTENTS IF YOU INTEND TO USE WORKING STORAGE PRIOR TO RESUMING THE INTERRUPTED PROGRAM. THE DATA SET CREATED BY THE NEWSPACE MUST BE ERASED JUST BEFORE THE RESTART, OR RESUMPTION WILL BE IMPOSSIBLE, AS PER THE ABOVE CONDITION.

(2) RESUMPTION IS ALSO IMPOSSIBLE IF THE ORIGINAL INTERRUPT OCCURRED WHILE THE RESIDENT MONITOR OR DISK I/O PROGRAM WAS IN CONTROL (EXCEPT FROM THE PRE- AND POST-OPERATIVE ERROR TRAPS OR "PROGRAM STOP" KEY SUBROUTINE). A SURE WAY OF CHECKPOINTING A RESUMABLE PROGRAM IS BY INCLUDING A WAIT OR PAUSE. THEN, WHEN THE PROGRAM WAITS AT THE PAUSE YOU CAN PRESS "INT REQ" AND IMMEDIATELY ISSUE THE CHECKPOINT COMMAND. YOU MAY THEN IMMEDIATELY ISSUE THE RESUME COMMAND, OR THE RESTART COMMAND WITH THE RESUME OPTION.

(3) YOU SHOULD NEVER ATTEMPT TO RESTART ANY PROGRAM WHICH USES THE WRITE IMMEDIATE FUNCTION OF EITHER THE DISK1 OR DISKN DISK I/O PROGRAMS. AS THE DISK ARMS CANNOT BE REPOSITIONED BEFORE PROGRAM EXECUTION IS RESUMED.

CLEAR OPTIONS

FUNCTION: RESET VARIOUS SEGMENTS OF THE SYSTEM CONTROL RECORD AREA PREVIOUSLY SET AS THE RESULT OF ISSUING ONE OF THE FIVE COMMANDS WHOSE COMMAND VERBS ARE USED AS THE OPTIONS TO THIS COMMAND.

PARAMETER:

OPTION(S): THE FOLLOWING ARE VALID OPTIONS:

FILES	DELETES ALL NAMEFILE CORRESPONDENCES CURRENTLY DEFINED.
OVERLAYS	DELETES ALL OVERLAY SPECIFICATIONS PREVIOUSLY GIVEN.
SUBSTITUTIONS	DELETES ALL CURRENT SUBROUTINE NAME SUBSTITUTIONS.
INCLUSIONS	DELETES ALL REFERENCES TO SUBROUTINES INCLUDED IN THE BUILDING OF LOAD MODULES, WHETHER CALLED OR NOT.
GRAPHICS	DELETES ALL GRAPHIC COMMAND OPTIONS PREVIOUSLY SET.
ALL	CLEARS THE ENTIRE SYSTEM CONTROL RECORD AREA. THIS IS EQUIVALENT TO SPECIFYING ALL OF THE OTHER FIVE OPTIONS TOGETHER.

EXAMPLE: ==> CLEAR (OVERLAYS, INCLUSIONS)

DELETE ANY REFERENCE TO SUBROUTINES PREVIOUSLY DEFINED AS EITHER BEING IN OVERLAY AREAS OR AS BEING INCLUDED IN THE BUILDING OF A CORE LOAD, WHETHER REFERENCED OR NOT.

NOTE: YOU MAY HAVE PREVIOUSLY DEFINED OVERLAY STRUCTURES OR THE LIKE FOR MORE THAN ONE MAINLINE PROGRAM. IF YOU WISH TO DELETE A REFERENCE TO ONLY ONE MAINLINE PROGRAM, YOU SHOULD NOT CLEAR THE AREA, BUT RATHER RE-ISSUE THE OVERLAY COMMAND, OR WHATEVER, SPECIFYING THE MAINLINE NAME YOU WISH TO DELETE FROM THE LIST ALONG WITH A NULL LIST OF SUBROUTINE NAMES TO BE OVERLAID. SEE THE OVERLAY COMMAND DESCRIPTION FOR A SPECIFIC EXAMPLE.

```
COBOL  INDSNAME  <OUTDSNAME>
                 <OPTIONS>  <'HEADING'>
```

FUNCTION: INVOKE THE IBM COBOL COMPILER TO PRODUCE AN OBJECT MODULE FROM A COBOL SOURCE PROGRAM.

PARAMETERS:

INDSNAME — INDSNAME NAMES THE DATA SET CONTAINING THE COBOL SOURCE PROGRAM. THE TYPE (.S) NEED NOT BE SPECIFIED. YOU MAY USE THE COPY FEATURE TO INCLUDE ADDITIONAL SOURCE CODE FROM A DIFFERENT DATA SET. .C (OR .CARDS) MAY BE SPECIFIED IF YOUR COBOL PROGRAM IS TO BE READ FROM CARDS. IF YOU WISH TO TYPE IN THE PROGRAM FROM THE KEYBOARD SPECIFY .T (OR .TERMINAL).

OUTDSNAME — OUTDSNAME IS THE NAME OF THE OBJECT MODULE PRODUCED. THE TYPE NEED NOT BE GIVEN: .O IS ASSUMED. IF OUTDSNAME IS OMITTED, NO OBJECT PROGRAM WILL BE PRODUCED (EQUIVALENT TO SPECIFYING .D IF PARAMETERS ARE TO FOLLOW OUTDSNAME). IF A NULL PARAMETER '()' IS SPECIFIED, AN OBJECT MODULE NAMED 'INDSNAME.O' WILL BE CATALOGUED. IF THE OBJECT MODULE NAME ALREADY EXISTS, THE OLD DATA SET WILL BE USED, PROVIDING THAT THE NEW OBJECT MODULE IS NO LARGER THAN THE OLD. IF YOU WISH TO SAVE THE OBJECT MODULE ON CARDS, SPECIFY .C.

OPTIONS — ANY OF THE STANDARD COBOL OPTIONS FROM THE FOLLOWING LIST MAY BE INCLUDED IN THE LIST OF OPTIONS:

LIST|NOLIST, DMAP|NODMAP, PMAP|NOPMAP,
STNO|NOSTNO, DUMP|NODUMP, XREF|NOXREF,
SUPX|NOSUPX, KP26|NOKP26, SUBR|NOSUBR,
EJCT|NOEJCT, DDMP, XMAP, #1442, #2501,
#1132, #1403, ORIGIN(/XXXX), S(NNN), CSUP

FOR CLARIFICATION OF THE MEANINGS OF EACH ONE OF THESE STANDARD COBOL OPTIONS, SEE THE '1130 COBOL OPERATIONS MANUAL' (SH20-0927). IN ADDITION, YOU MAY SPECIFY HERE IF YOU WISH PRINTED OUTPUT TO BE ROUTED TO THE TERMINAL INSTEAD OF TO THE PRINCIPAL PRINTER.

'HEADING' — IF GIVEN, WILL BE PRINTED AT THE TOP OF EACH PAGE OF PRINTED OUTPUT.

EXAMPLE: ==> COBOL ACCTR () (LIST,DMAP,NOSTNO) &
'ACCOUNTS RECEIVABLE'

COMPILE THE SOURCE PROGRAM ACCTR AND NAME THE OBJECT MODULE, IF ONE IS SUCCESSFULLY PRODUCED, ACCTR.O. A LISTING, AS WELL AS A MAP OF THE DATA DIVISION AND SUPPRESSION OF STATEMENT NUMBERING, IS INDICATED BY THE OPTION LIST. 'ACCOUNTS RECEIVABLE' WILL BE PRINTED AT THE TOP OF EACH PAGE OF PRINTED OUTPUT.

NOTE: THE OPTIONS SPECIFIED OVERRIDE THOSE SPECIFIED IN THE SOURCE PROGRAM OPTION CONTROL CARD IF THE SAME FEATURE IS MENTIONED IN BOTH LOCATIONS. THUS, IF LIST IS SPECIFIED WITHIN THE SOURCE PROGRAM, AND NOLIST IS GIVEN AS AN OPTION TO THE COMMAND, THE NOLIST WILL PREVAIL.

IF TWO CONFLICTING OPTIONS ARE GIVEN WITHIN THE COMMAND OPTION LIST (E.G. STNO AND NOSTNO), THE RESULTS WILL BE THE NO CONDITION.

IF A FEATURE IS MENTIONED IN ONE PLACE BUT NOT THE OTHER, THE MENTIONED SETTING WILL PREVAIL.

IF NO MENTION OF AN OPTION IS MADE EITHER IN THE COMMAND OR WITHIN THE SOURCE PROGRAM, THE DEFAULT CONDITION WILL PREVAIL.

COMMA

FUNCTION: FORMAT AND PRINT THE CONTENTS OF VARIOUS SYSTEM LOCATIONS IN THE LOW-CORE 'COMMUNICATIONS AREA' OF A PROGRAM THAT HAS BEEN INTERRUPTED DURING EXECUTION BY PRESSING THE 'INT REQ' KEY.

THIS COMMAND IS OF USE PRIMARILY IN THE DEBUGGING OF SYSTEM PROGRAMS AND THOSE USER PROGRAMS WHOSE EXECUTION MIGHT DEPEND UPON NON-STANDARD USE OF THE DISK MONITOR SYSTEM.

CONTINUE <OPTION>

FUNCTION: CONTINUE EXECUTE-MODE OR CARD-INPUT-MODE PROCESSING OF CYTOS COMMANDS PREVIOUSLY SUSPENDED BY PRESSING THE INT REQ KEY.

OPTION: THE LOGGING OF PROMPTS AND INTERPRETED COMMANDS MAY BE ALTERED BY SPECIFYING ONE OF THE FOLLOWING OPTIONS.

HERE — PRINTED OUTPUT ROUTED TO THE TERMINAL AS UNDER CONVERSATIONAL MODE.

PRINTER — PRINTED OUTPUT ROUTED TO THE PRINCIPAL SYSTEM PRINTER.

NOPRINT — PRINTING OF CYTOS PROMPTS AND INTERPRETED CYTOS COMMANDS TO BE SUPPRESSED; ONLY ERROR MESSAGES AND NORMAL INFORMATION PRINTED BY THE COMMANDS THEMSELVES ARE TO BE PRINTED AT THE TERMINAL.

EXAMPLE: ==> CONTINUE PRINTER

EXECUTION OF CYTOS COMMANDS FROM A DISK DATA SET WILL BE RESUMED FROM THE POINT OF PREVIOUS INTERRUPTION, AND ALL PRINTED OUTPUT WILL HENCEFORTH BE ROUTED TO THE PRINCIPAL PRINTER INSTEAD OF TO THE TERMINAL.

NOTES: (1) THE CONTINUE COMMAND ALLOWS YOU TO SWITCH BACK AND FORTH BETWEEN CONVERSATIONAL AND EXECUTE MODES FOR GREATER PROCESSING FLEXIBILITY. CONVERSATIONAL MODE MAY BE ENTERED FROM EXECUTE MODE EITHER BY PRESSING INT REQ (AND THUS ABORTING THE CYTOS COMMAND THEN BEING EXECUTED) OR BY EMBEDDING A KEYBOARD COMMAND AT THE APPROPRIATE PLACE(S) IN THE EXECUTE DATA SET. OPERATION UNDER CARD-INPUT MODE IS IDENTICAL TO THAT UNDER EXECUTE MODE.

(2) IF NO OPTION IS SUPPLIED, FURTHER PRINTING WILL BE ROUTED TO THE SAME DEVICE USED PRIOR TO ENTERING CONVERSATIONAL MODE.

COPY (OLDDSNAME,...) (NEWDSNAME,...)

FUNCTION: MAKE A COPY OF EACH OF ONE OR MORE DATA SETS.

ALIAS: C

PARAMETERS:

OLDDSNAME — DATA SETS TO BE COPIED FROM; S, O, L, OR * MAY BE SPECIFIED; OLDDSNAME MAY BE ANY VALID CYTOS DATA SET NAME.

NEWDSNAME — DATA SET TO BE COPIED INTO; IF TYPE IS NOT GIVEN, IT DEFAULTS TO THE TYPE OF THE CORRESPONDING NAME IN THE LIST OF OLDDSNAMES. IF NEWDSNAME ALREADY EXISTS, NO COPY WILL TAKE PLACE.

EXAMPLE: ==> COPY (PROG.*,DATA) (VER5.*,DATA5)

CREATE NEW DATA SETS CALLED VER5.S, VER5.O, VER5.L, AND DATA5.S, THEN COPY INTO THEM THE CURRENT CONTENTS OF THE DATA SETS PROG.S, PROG.O, PROG.L, AND DATA.

NOTE: TYE TYPES OF CORRESPONDING ITEMS IN THE TWO LISTS MUST ALWAYS AGREE; OTHERWISE, ALL COPYING WILL BE TERMINATED AND CYTOS WILL PROMPT FOR THE NEXT COMMAND.

YOU MAY COPY FROM AND INTO DATA SETS BELONGING TO OTHER USERS SO LONG AS THEY HAVE GIVEN YOU PERMISSION TO READ FROM AND COPY ANY OLDSNAME, AND TO WRITE INTO ANY NEWDSNAME.

DCOM

FUNCTION: FORMAT AND PRINT THE CONTENTS OF THE 'DISK COMMUNICATIONS AREA' ON THE CURRENTLY DEFINED MASTER CARTRIDGE, I.E., THE CARTRIDGE FROM WHICH CYTOS IS CURRENTLY RUNNING.

NOTE: THIS COMMAND IS OF USE PRIMARILY IN DETERMINING THE LOCATION OF VARIOUS LIBRARIES AND SYSTEM AREAS ON THE CURRENTLY MOUNTED VOLUMES. FOR AN EXPLANATION OF THE MEANING OF EACH OF THE SYMBOLIC LOCATIONS WITHIN THE DISK COMMUNICATIONS AREA, SEE '1130 DISK MONITOR SYSTEM, VERSION 2, PROGRAMMER AND OPERATOR'S GUIDE,' APPENDIX G.

DESCRIBE (DSNAME, ...)

FUNCTION: PROVIDE INFORMATION ABOUT ONE OR MORE DATA SETS.

ALIAS: D

PARAMETER:

DSNAMES — A LIST OF ONE OR MORE DATA SET NAMES ABOUT WHICH INFORMATION IS DESIRED. ANY DATA SET TYPE MAY BE SPECIFIED. IF YOU WISH TO GET INFORMATION ABOUT ALL TYPES OF A DATA SET, YOU MAY SAY 'DSNAME.*'.

OUTPUT:
1. DATA SET TYPE (.S, .O, .L);
2. VOLUME ON WHICH DATA SET IS CATALOGUED;
3. LOCATION OF THE DATA SET ON THE VOLUME (THE 'DISK BLOCK ADDRESS'); AND
4. THE SIZE OF THE DATA SET (IN DISK BLOCKS).

DUMP (DSNAME, ...) <HERE>

FUNCTION: PRINT THE CONTENTS OF ONE OR MORE DATA SETS IN HEXADECIMAL FORMAT.

PARAMETERS:

DSNAMES — A LIST OF DATA SETS TO BE DUMPED. ANY TYPE (.S, .O, OR .L) MAY BE DUMPED, AND IF YOU WISH TO DUMP ALL TYPES OF A DATA SET NAME YOU MAY SPECIFY DSNAME.*.

HERE — THE PRINTED HEXADECIMAL OUTPUT WILL BE AT THE TERMINAL IF HERE IS SPECIFIED; OTHERWISE, THE DUMP WILL TAKE PLACE ON THE PRINCIPAL PRINT DEVICE.

EXAMPLE: ==> DUMP (NTWRK.*, PARKR.TYPE6.L)

DUMP TO THE PRINCIPAL PRINTER IN HEXADECIMAL FORMAT ANY OF THE FOLLOWING DATA SETS THAT ARE CURRENTLY CATALOGUED:

```
NTWRK.S
NTWRK.O
NTWRK.L
PARKR.TYPE6.L
```

```
EDIT <DSNAME> <ATTRIBUTE> <RECORD_LENGTH>
```

FUNCTION: EDIT AN EXISTING SOURCE DATA SET, OR CREATE AND ENTER INFORMATION INTO A NEW DATA SET. THIS IS THE USUAL WAY FOR YOU TO CREATE A SOURCE DATA SET FROM THE TERMINAL.

ALIAS: E

PARAMETERS:

DSNAME — THE NAME OF THE DATA SET TO BE EDITED. IF OMITTED, OR IF THE NAME DOES NOT ALREADY EXIST IN YOUR CATALOGUE, THEN THE EDITOR ASSUMES THAT YOU WANT TO CREATE A NEW DATA SET, AND WILL AUTOMATICALLY ENTER 'INSERT' MODE. IF THE DATA SET IS ALREADY CATALOGUED, THEN THE EDITOR WILL ASSUME THAT THE DATA SET EXISTS AND IS TO BE MODIFIED.

ATTRIBUTE — DATA SET ATTRIBUTE IS USED BY THE EDITOR TO ESTABLISH INITIAL TAB SETTINGS APPROPRIATE TO THE SOURCE TEXT BEING EDITED. THE POSSIBLE ATTRIBUTES ARE:

FORTRAN|ASSEMBLER|RPG|COBOL|ASM360|DATA

IF, TOGETHER WITH THE ATTRIBUTE IN A COLLECTION, YOU SPECIFY QMODE, THE EDITOR WILL ENTER Q-MODE AND ALLOW YOU TO SET VARIOUS OPTIONS AND UTILITY PARAMETERS BEFORE ANY EDITING IS STARTED.

RECORD_LENGTH — THE NUMBER OF CHARACTERS PER SOURCE RECORD. THE DEFAULT IS 80 CHARACTERS PER RECORD.

EXAMPLE: ==> EDIT PROGA (FORTRAN, QMODE)

EDIT THE OLD (EXISTING) FORTRAN PROGRAM WHOSE NAME IS PROGA. ENTER THE EDITOR IN Q-MODE TO ALLOW INITIAL EDITING PARAMETERS TO BE SET.

REFERENCE: SEE SECTION 5 OF THIS MANUAL FOR A COMPLETE TUTORIAL ON THE OPERATIONS OF THE CYTOS EDITOR.

INPUT: THE EDITOR PROMPTS FOR INPUT IN THREE DIFFERENT WAYS: I=> IN INSERT MODE, E=> IN EDIT MODE, AND Q=> IN Q-MODE. THE COMMANDS AVAILABLE IN THE EDITOR ARE EXPLAINED IN THE "EDIT TUTORIAL" SECTION OF THIS MANUAL; THEY ARE LISTED BELOW.

```
    AFTER 'PATTERN' 'REPLACEMENT' <NN> <MM>
(A) ANCHOR 'PATTERN' <NN>
    ATTN
    BEFORE 'PATTERN' 'REPLACEMENT' <NN> <MM>
(B) BOTTOM
    CAPITALIZE 'PATTERN' <NN>
(C) CHANGE 'PATTERN' 'REPLACEMENT' <NN> <MM>
    CWALLS (N1 N2 ... N10)
    DEFINE PROC_NAME (COMMAND, ...)
(D) DELETE <NN>
    DISPLAY
    DO (COMMAND, ...) <NN>
(E) END
    EXPLAIN (TERM1 TERM2 ... TERMN)
(F) FILE <DSNAME> <RECORD_LENGTH>
    FOLD 'PATTERN' <NN>
    INFL DSNAME <RECORD_LENGTH> <START> <END>
(I) INSERT <'LINE'> <NN>
    LABEL <NN> <'GANG'> <NEW>
    LL <NN>
(L) LOCATE 'PATTERN' <NN>
(N) NEXT <NN>
(O) OVERLAY 'REPLACEMENT' <NN>
    PL <NN> <MM>
(P) PRINT <NN>
    PWALLS (N1 N2 ... N10)
(Q) QUIT
(R) REPLACE 'LINE' <NN>
    SAVE DSNAME
(S) SET STR_VAR|SWITCH|CHAR_FUNC 'VALUE'
    TABS (N1 N2 ... N10)
(T) TOP
(U) UPDATE <DSNAME> <RECORD_LENGTH>
    USE DSNAME
    UTILITY DSNAME
(W) WHATS 'STRING'
    'PROCEDURE_NAME'
```

```
-----------------------------------------------------------
|                                                         |
|              EJECT  <NN>  <'HEADING'>                   |
|                                                         |
-----------------------------------------------------------
```

FUNCTION: EJECT NN PAGES ON THE PRINCIPAL PRINTER.

PARAMETERS: NN SPECIFIES THE NUMBER OF PAGES. IF OMITTED OR NULL, THE FORMS WILL BE POSITIONED AT THE TOP OF THE NEXT PAGE.

A HEADING MAY BE PRINTED AT THE TOP OF EACH PAGE.

```
-----------------------------------------------------------
|                                                         |
|                ERASE  (DSNAME, ...)                     |
|                                                         |
-----------------------------------------------------------
```

FUNCTION: ERASE AND UNCATALOGUE ONE OR MORE DATA SETS.

ALIAS: DELETE

PARAMETERS:

DSNAME YOU MAY SPECIFY ANY TYPE OF DATA SET, INCLUDING '*' WHICH WILL CAUSE ALL TYPES FOR THE GIVEN NAME TO BE ERASED.

EXAMPLES: ==> ERASE (SIGFRIED.COEF.O, CODAT.S(3012))

WOULD CAUSE TWO DATA SETS TO BE ERASED: SIGFRIED'S OBJECT MODULE CALLED COEF, ASSUMING THAT SIGFRIED HAS GIVEN YOU PERMISSION TO ERASE HIS DATA SETS, AND YOUR OWN DATA SET CALLED CODAT, ASSUMING THAT A DATA SET OF THAT NAME CURRENTLY CAN BE FOUND ON THE VOLUME 3012 AND THAT THIS VOLUME ACUTALLY BELONGS TO YOU.

```
EXECUTE  DSNAME  <OPTION>  <COLUMNS>
```

FUNCTION: ACCEPT CYTOS COMMANDS FROM THE SPECIFIED DATA SET AND INTERPRET THEM SEQUENTIALLY JUST AS IF THEY HAD BEEN ENTERED FROM THE TERMINAL.

ALIAS: XEQ

PARAMETERS:

DSANME THE NAME OF A SOURCE DATA SET IN WHICH CYTOS COMMANDS HAVE BEEN PREVIOUSLY STORED ONE PER 80-CHARACTER LOGICAL RECORD. CONTROL IS RETURNED TO THE KEYBOARD AFTER THE LAST COMMAND IN THE DATA SET HAS BEEN PROCESSED.

OPTION ONE OF THE FOLLOWING OPTIONS MAY BE SPECIFIED TO INDICATE THE DISPOSITION OF THE PRINTED OUTPUT FROM THE COMMANDS IN THE EXECUTED DATA SET:

HERE ALL PRINTED OUTPUT TO BE ROUTED AS UNDER CONVERSATIONAL MODE.

PRINTER PRINTED OUTPUT, INCLUDING CYTOS PROMPTS AND INTERPRETED CYTOS COMMANDS, TO BE ROUTED TO THE PRINCIPAL SYSTEM PRINTER.

NOPRINT PRINTING OF CYTOS PROMPTS AND INTERPRETED COMMANDS TO BE SUPPRESSED; ONLY THE NORMAL PRINTED OUTPUT OF CYTOS COMMANDS, INCLUDING ERROR MESSAGES, TO BE PRINTED TO THE DEVICE NORMALLY USED IN CONVERSATIONAL MODE FOR SUCH MESSAGES.

COLUMNS A DECIMAL COLUMN NUMBER INDICATING THE END OF THE FIELD IN WHICH CYTOS COMMANDS MAY BE FOUND. THE DEFAULT UNDER EXECUTE MODE IS 80, OR THE ENTIRE RECORD. UNDER CARD-INPUT MODE, THE DEFAULT IS 72, AS IT IS MORE LIKELY THAT 8 CHARACTER SEQUENCE NUMBERS MAY BE PLACED AT THE END OF EACH COMMAND RECORD.

EXAMPLE: ==> EXECUTE TRYIT PRINTER 72

THE DATA SET TRYIT.S MIGHT HAVE THE FOLLOWING SET OF CYTOS COMMANDS TO TEST A FORTRAN PROGRAM:

```
ERASE (PROG.O, PROG.L)
FORTRAN PROG PROG (LIST ATRACE #1403)
LOADGO PROG MAP
DATAX (REPLY TO FIRST WHATFILE PROMPT)
# (REPLY TO SECOND WHATFILE PROMPT)
PRINT DATAX
REMOVE DATAX
.RESET (CONTINUES HERE ON COMPILATION ERROR)
DESCRIBE PROG.*
.END
```

ALL PRINTING WILL BE DONE ON THE PRINCIPAL PRINTER; CYTOS COMMANDS OCCUPY ONLY THE FIRST 72 CHARACTERS OF EACH CONTROL RECORD.

NOTES: (1) SEE SECTION 8 FOR AN EXPLANATION OF THE PSEUDO-COMMANDS .RESET, .ATTENTION, AND .END THAT MAY BE USED IN EXECUTE MODE TO SIMULATE VARIOUS CONVERSATIONAL ATTRIBUTES.

(2) EXECUTE COMMANDS MAY BE NESTED TO A DEPTH OF TWELVE (12). WHEN AN EXECUTE COMMAND IS ENCOUNTERED WITHIN AN EXECUTE DATA SET, THE STATUS OF THE OUTER EXECUTE IS AUTOMATICALLY SAVED, AND THEN RESTORED UPON REACHING THE END OF THE INNER EXECUTE PROCESS. SEQUENCE NUMBERS MAY BE USED TO INDICATE THE LEVEL OF NESTING IN THE PRINTED OUTPUT.

(3) CYTOS EDITOR COMMANDS MAY BE ENTERED MORE THAN ONE PER RECORD. SEE THE EDITOR TUTORIAL, SECTION 5.E, FOR DETAILS.

(4) COMMAND SYNTAX IN EXECUTE MODE IS IDENTICAL TO THAT IN CONVERSATIONAL MODE: COMMANDS MAY BE CONTINUED ONTO SUCCESSIVE RECORDS BY USING THE LOGICAL CONTINUATION CHARACTER ('&'), ETC.

EXPLAIN (TERM1, TERM2, ..., TERM_N)

FUNCTION: PRINT DESCRIPTIONS OF CYTOS COMMANDS, SYSTEM ERROR CODES, LANGUAGE PROCESSOR DIAGNOSTICS, AND CURRENT CYTOS REVISIONS AND RESTRICTIONS.

ALIAS: X

PARAMETERS: AS MANY TERMS AS DESIRED MAY BE GIVEN TO THE EXPLAIN COMMAND. IN ADDITION, SOME TERMS MAY BE SUBSCRIPTED, INDICATING THAT ONLY A PORTION OF A FULL DESCRIPTION IS DESIRED: SEE EXAMPLES 3 AND 5 BELOW.

EXAMPLES: (1) ==> EXPLAIN EXPLAIN

DESCRIBE THE EXPLAIN COMMAND ITSELF, AND THE VARIOUS TERMS THAT CAN BE EXPLAINED.

(2) ==> EXPLAIN CYTOS.COMMANDS

LIST THE CURRENTLY IMPLEMENTED CYTOS COMMANDS.

(3) ==> EXPLAIN FORTRAN.XTRACE

DESCRIBE THE XTRACE OPTION OF THE FORTRAN COMMAND.

(4) ==> EXPLAIN (PD56, D93, READ)

DESCRIBE THE CYTOS ERROR MESSAGE WITH THE KEY PD56, THE DMS ERROR MESSAGE D93, AND THE CYTOS READ COMMAND.

(5) ==> EXPLAIN (LOADGO.OPTIONS, LINKEDIT.OPTIONS)

EXPLAINS THE OPTIONS TO THE LOADGO AND LINKEDIT COMMANDS.

NOTE: THE EXPLAIN COMMAND REFERS TO A SPECIAL DATA SET CYTXP WHICH MUST BE CATALOGUED AT THE TIME THE COMMAND IS GIVEN.

FIXLIB CARTRIDGE_ID SIZE

FUNCTION: ESTABLISH A FIXED AREA ON A REMOVABLE DISK VOLUME.

PARAMETERS:

CARTRIDGE — THE CARTRIDGE_ID OF THE VOLUME ON WHICH THE LIBRARY IS TO BE ESTABLISHED. THE VOLUME MUST BE MOUNTED AT THE TIME THE COMMAND IS ISSUED FOR THE LIBRARY TO BE ESTABLISHED.

SIZE — THE NUMBER OF CYLINDERS OF LIBRARY SPACE TO BE CREATED. IF A FIXED AREA ALREADY EXISTS ON THE SPECIFIED VOLUME, THIS SIZE REPRESENTS AN INCREMENT (IF POSITIVE) OR DECREMENT (IF NEGATIVE) TO THE CURRENT SIZE OF THE AREA. IF THE DECREMENT IS GREATER THAN THE CURRENT SIZE, THE LIBRARY WILL BE ENTIRELY REMOVED FROM THE CARTRIDGE.

EXAMPLE: FIXLIB 1005 20

WOULD ESTABLISH A FIXED AREA LIBRARY (ASSUMING THAT ONE DID NOT ALREADY EXIST) ON THE VOLUME WHOSE CARTRIDGE ID IS 1005 WITH AN INITIAL SIZE OF 20 CYLINDERS.

NOTE: WHEN ESTABLISHING A FIXED AREA, THE ACTUAL SIZE WILL BE ONE CYLINDER LESS THAN SPECIFIED IN THE COMMAND. THE FIRST CYLINDER WILL BE USED FOR THE LIBRARY DIRECTORY (FLET).

FLUSH

FUNCTION: FEED CARDS THROUGH THE CARD READER UP TO AND INCLUDING THE NEXT CONTROL CARD ('//' IN COLUMNS 1 AND 2). THIS COMMAND ALLOWS YOU TO BYPASS A CARD DECK, AS FOR INSTANCE WHEN A COMPILATION TO CARDS HAD ERRORS AND THE OUTPUT DECK CANNOT BE PUNCHED ON THE 1442-6 OR -7 CARD READ/PUNCH UNIT.

```
FORTRAN  INDSNAME  <OUTDSNAME>
    <OPTIONS>  <'HEADING'>
```

FUNCTION: INVOKE THE IBM OR DNA FORTRAN COMPILER TO COMPILE A FORTRAN SOURCE PROGRAM.

ALIASES: FOR, F

PARAMETERS:

INDSNAME THE NAME OF THE SOURCE DATA SET CONTAINING THE FORTRAN STATEMENTS TO BE COMPILED. THE TYPE DEFAULTS TO .S IF OMITTED. IF YOU WISH TO TYPE IN THE PROGRAM FROM THE KEYBOARD, YOU MAY SPECIFY .T (OR .TERMINAL); OR IF THE SOURCE PROGRAM IS IN CARD FORM, YOU MAY SAY .C (OR .CARDS). YOU MUST ISSUE THE FORTRAN COMMAND FOR EACH MAINLINE PROGRAM OR SUBROUTINE YOU WANT TO COMPILE; THAT IS, ONLY THE FIRST PROGRAM IN ANY DATA SET WILL BE COMPILED.

OUTDSNAME THE NAME OF THE DATA SET TO CONTAIN THE OBJECT MODULE (I.E., THE RESULTS OF THE COMPILATION). THE TYPE NEED NOT BE GIVEN; .O IS ASSUMED. IF OUTDSNAME IS OMITTED, NO OBJECT MODULE WILL BE PRODUCED; IF A NULL () PARAMETER IS SPECIFIED, AN OBJECT MODULE NAMED INDSNAME.O WILL BE CATALOGUED. IF OUTDSNAME.O ALREADY EXISTS, IT WILL BE REUSED, SO LONG AS THE NEW VERSION IS NO LARGER THAN THE OLD. IF YOU WISH TO SAVE THE OBJECT MODULE ON CARDS, YOU MAY SPECIFY .C; AND IF YOU WISH NO OBJECT MODULE AT ALL, YOU MAY SPECIFY .D (OR .DUMMY).

OPTIONS: THE FOLLOWING OPTIONS ARE ALLOWED AND WILL HAVE THE INDICATED EFFECTS:

LIST — LIST THE SOURCE PROGRAM.

NOLIST — SUPPRESS THE LISTING OF THE SOURCE PROGRAM, THE SYMBOL TABLE, THE SUBROUTINE CROSS REFERENCE, AND ERROR TEXTS.

XREF — LIST SUBROUTINE NAMES.

SYMBOLS — LIST SYMBOL NAMES.

TEXTS — LIST ERROR TEXTS (EXPLANATIONS OF C-ERRORS) AFTER THE COMPLETION OF AN UNSUCCESSFUL COMPILATION. THIS OPTION IGNORED IF THE IBM COMPILER IS ON THE SYSTEM.

ALL — LIST EVERYTHING (EQUIVALENT TO LIST + XREF + SYMBOLS + TEXTS).

HERE	THE LISTING WILL BE PRINTED AT YOUR TERMINAL, INSTEAD OF ON THE PRINCIPAL PRINT DEVICE.
ATRACE	INCLUDE THE ASSIGNMENT TRACE FEATURE.
XTRACE	INCLUDE THE TRANSFER TRACE FEATURE.
NOTRACE	SUPPRESS THE ASSIGNMENT AND TRANSFER TRACE FEATURES WHEN COMPILING THE SOURCE PROGRAM.
EXTEND	INDICATES EXTENDED PRECISION ARITHMETIC.
STANDARD	STANDARD PRECISION (THE DEFAULT).
LONG (OR USASI)	CONFORM TO USASI STANDARD TWO-WORD (32 BIT) INTEGERS IN THE COMPILED PROGRAM.
NOREAL	OVERRIDE THE PRECISION AND TYPE IMPLIED BY THE SUBROUTINE NAME SO AS TO AVOID CONFLICTS WITH ALL MAINLINE PROGRAMS.
NOINTEGER	OVERRIDE INTEGER TYPING OF SUBROUTINE SO AS TO AVOID CONFLICTS WHEN CALLED IN NON-INTEGER CONTEXT.
NOIOCS	SUPPRESS ALL IOCS DEVICE OPTIONS EXCEPT THOSE EXPLICITLY PROVIDED IN THE LIST OF OPTIONS IN THE FORTRAN COMMAND ITSELF.
RESET	SUPPRESS ALL NON-IOCS OPTIONS EXCEPT THOSE EXPLICITLY PROVIDED IN THE LIST OF OPTIONS IN THE FORTRAN COMMAND ITSELF.
SHORT	INDICATES ONE-WORD INTEGERS.
ORIGIN(N)	INDICATES THAT THE ORIGIN OF THE PROGRAM IS TO BE LOCATED ABSOLUTELY AT LOCATION N. THE NUMBER N MAY BE EITHER OF THE FORM 'DDDDD', WHERE EACH 'D' REPRESENTS ONE DECIMAL DIGIT, OR '/HHHH', WHERE EACH 'H' REPRESENTS ONE HEXADECIMAL DIGIT.
DEVICES	ANY ONE OR MORE OF THE FOLLOWING INPUT/OUTPUT DEVICE SUBROUTINES MAY BE INCLUDED IN THE LIST OF OPTIONS: CARD, KEYBOARD, TYPEWRITER, PAPERTAPE, DISK, UDISK, PLOTTER, #1132, #1403, #1442, #2501

FDISK — FORMATTED DISK I/O. THIS OPTION IGNORED IF THE IBM COMPILER IS ON THE SYSTEM.

AS IN BATCH MODE, THE DEVICES CARD AND #1442 ARE MUTUALLY EXCLUSIVE.

NOTE THAT IF MORE THAN ONE OPTION IS GIVEN, THE OPTIONS MUST BE ENCLOSED IN PARENTHESES.

'HEADING' — ANY VALID HEADING OF UP TO 66 CHARACTERS MAY BE PROVIDED TO BE PRINTED AT THE TOP OF EACH PAGE OF PRINTED OUTPUT. THE ENTIRE HEADING MUST BE ENTERED ON ONE LINE (I.E., THE '&' CONTINUATION CHARACTER HAS NO EFFECT BETWEEN HEADING QUOTES).

REFERENCE:

```
==> FORTRAN KING.CVFIT.S CURVE.O (ALL,KEYBOARD, &
&=> TYPEWRITER) 'KING''S CURVE FITTING PROGRAM'
```

WILL CAUSE THE INDICATED SOURCE PROGRAM TO BE COMPILED AND THE RESULTING OBJECT MODULE SAVED IN THE DATA SET CURVE.O; A COMPLETE LISTING WILL BE GIVEN ON THE PRINCIPAL PRINTER, EACH PAGE HAVING THE HEADING GIVEN AS THE FOURTH PARAMETER TO THE COMMAND.

NOTES:

(1) ANY OPTIONS SPECIFIED IN THE FORTRAN COMMAND WILL BE ADDED TO THOSE ALREADY SPECIFIED ON *OPTION CARDS AT THE BEGINNING OF THE FORTRAN SOURCE PROGRAM. THE ORIGIN LOCATION AND 'HEADING' WILL OVERRIDE CORRESPONDING VALUES IN THE SOURCE PROGRAM. IF YOU WISH TO DELETE ANY OPTIONS FROM THE SOURCE PROGRAM ITSELF, YOU SHOULD USE THE APPROPRIATE 'NO' OPTION.

(2) YOU MUST ISSUE A SEPARATE FORTRAN COMMAND FOR EACH MAINLINE OR SUBROUTINE YOU WISH TO COMPILE; THAT IS, ONLY THE FIRST SOURCE PROGRAM IN ANY SOURCE DATA SET WILL BE RECOGNIZED BY THE FORTRAN COMPILER.

```
GRAPHICS  PROGRAM_NAME  <OPTIONS>
```

FUNCTION: PROVIDES SELECTIVE INCLUSION OF VARIOUS COMPONENTS OF THE GRAPHICS SUBROUTINE PACKAGE WHEN BUILDING LOAD MODULES.

ALIAS: G, G2250

PARAMETERS:

PROG_NAME — THE NAME OF THE MAINLINE PROGRAM WITH WHICH THE VARIOUS PORTIONS OF THE GRAPHICS SUBROUTINE PACKAGE ARE TO BE INCLUDED.

OPTIONS — A COLLECTION OF ONE OR MORE OF THE FOLLOWING KEYWORDS MAY BE SPECIFIED:

NOSTROKE — DO NOT INCLUDE CHARACTER STROKE SUBROUTINE.

LOWERCASE — INCLUDE SUBROUTINE FOR UPPER CASE, LOWER CASE AND ALL SPECIAL CHARACTERS.

IF NEITHER NOSTROKE NOR LOWERCASE IS SPECIFIED, THE LOAD MODULE WILL INCLUDE THE SUBROUTINE FOR UPPER CASE CHARACTERS AND SPECIAL CHARACTERS ONLY.

SCISSOR — INCLUDE SCISSORING SUBROUTINE.

EXPANSION — INCLUDE THE ICA AREA EXPANSION SUBROUTINE.

INDEX — INCLUDE THE INDEX CONTROLLED ENTITY SUBROUTINE.

LEVEL — INCLUDE LEVEL CONTROLLED DIRECT ENTITY SUBROUTINE.

ALL — EQUIVALENT TO LOWERCASE + SCISSOR + EXPANSION + INDEX + LEVEL

APPEND — ADD OPTIONS TO THOSE ALREADY DEFINED FOR THIS PROGRAM. IF OMITTED, ANY PREVIOUS SPECIFICATIONS WILL BE IGNORED.

LIST — LIST ALL GRAPHICS OPTIONS CURRENTLY DEFINED.

EXAMPLE: ==> GRAPHICS DRAW (LOWERCASE,SCISSOR)

IN BUILDING THE MAINLINE PROGRAM DRAW, INCLUDE THE FULL CHARACTER SET SUBROUTINE, AS WELL AS THE SCISSORING SUBROUTINE.

NOTES: (1) TO REVOKE A PREVIOUSLY DEFINED SET OF GRAPHICS FEATURES FOR THE BUILDING OF A SPECIFIC MAINLINE PROGRAM, ISSUE THE GRAPHICS COMMAND FOR THE SAME PROGRAM_NAME, BUT OMIT THE SECOND PARAMETER.

(2) TO REVOKE GRAPHICS DEFINITIONS ON ALL MAINLINE PROGRAMS, ISSUE THE CLEAR GRAPHICS COMMAND.

INCLUDE PROGRAM_NAME <(SUBROUTINE_NAME, ...)>
<LIST,APPEND>

FUNCTION: INCLUDE ONE OR MORE SUBROUTINES IN A LOAD MODULE, EVEN THOUGH EACH MAY NOT BE REFERENCED BY EITHER THE MAINLINE PROGRAM OR BY ANY OTHER SUBROUTINE REFERENCED BY THE MAINLINE PROGRAM.

ALIAS: I, NOCAL

PARAMETERS:

PROG_NAME THE NAME OF THE MAINLINE PROGRAM WITH WHICH THE VARIOUS SUBROUTINES ARE TO BE INCLUDED WHEN A LOAD MODULE IS BUILT BY THE CORE LOAD BUILDER.

SUB_NAMES ONE OR MORE SUBROUTINES, IN OBJECT MODULE FORM, TO BE INCLUDED WITH THE MAINLINE PROGRAM. THE SUBROUTINES MUST BE YOURS; THAT IS, YOU MAY NOT REFER TO ANY SUBROUTINE BY MODIFYING THE NAME WITH ITS OWNER. IF YOU WISH TO INCLUDE SOMEONE ELSE'S SUBROUTINES, YOU MUST FIRST COPY HIS SUBROUTINE INTO YOUR OWN LIBRARY. YOU NEED NOT MODIFY THE SUBROUTINE NAME WITH THE TYPE, FOR .O IS ASSUMED.

OPTIONS: THE THIRD PARAMETER MAY BE ONE OR BOTH OF THE FOLLOWING:

APPEND IF YOU WISH TO ADD THIS GROUP OF SUBROUTINE NAMES TO THOSE ALREADY ASSOCIATED WITH THE PROGRAM_NAME SPECIFIED. IF APPEND IS NOT GIVEN, ALL PREVIOUS INCLUSIONS FOR THE SPECIFIED MAINLINE PROGRAM WILL BE OVERRIDDEN.

LIST — CAUSES A LIST OF ALL SUBROUTINES CURRENTLY DEFINED AS BEING INCLUDED IN THE BUILDING OF LOAD MODULES TO BE PRINTED AT THE TERMINAL. THIS OPTION ALLOWS YOU TO EXAMINE THE ENTIRE DEFINITION TO MAKE SURE YOU HAVE INCLUDED EVERYTHING YOU NEED.

EXAMPLE:

```
==> INCLUDE PROGB (SRA,SRB.0,SRC) (APPEND.LIST)
```

WOULD CAUSE THE SUBROUTINES SRA, SRB, AND SRC TO BE INCLUDED IN ANY LOAD MODULE BUILT FOR THE MAINLINE PROGRAM PROGB; THE NAMES WOULD BE ADDED TO ANY PREVIOUSLY DEFINED TO BE INCLUDED WITH PROGB, AND A LIST OF ALL CURRENTLY DEFINED NAMES WOULD BE PRINTED:

```
SUBROUTINES INCLUDED BUT NOT CALLED:
MAINLINE - SUBROUTINES ...
PROGB      SRA   SRB   SRC   PAUSE   ILS02
PROGA      PAUSE ILS02 PRNTZ
```

NOTES:

(1) THE INCLUDE COMMAND MSUT BE GIVEN BEFORE THE LOADGO OR LINKEDIT COMMAND IS ISSUED TO ACTUALLY BUILD THE LOAD MODULE.

(2) IF A LOAD MODULE IS TO BE BUILT USING THE LINKEDIT COMMAND, THE 'PROGRAM_NAME' PARAMETER MUST BE NULL () BECAUSE AT THE TIME THE LOAD MODULE IS BEING BUILT, THE OBJECT MODULE IS IN SYSTEM WORKING STORAGE, AND THEREFORE HAS NO NAME ASSOCIATED WITH IT.

(3) IF YOU WISH TO DISASSOCIATE ALL SUBROUTINE NAMES FROM A GIVEN PORGRAM_NAME, YOU MAY ISSUE THE INCLUDE COMMAND WITH A NULL LIST OF SUBROUTINES:

```
==> INCLUDE PROGB ()
```

JOIN USERID PROJID CARTRIDGE_ID <ALL>

FUNCTION: ESTABLISH OWNERSHIP OF A CYTOS SYSTEM VOLUME OR DMS SYSTEM OR SATELLITE VOLUME. YOU WILL BE PROMPTED FOR THE NEW PASSWORD.

ALIAS: J

PARAMETERS:

USERID — THE PERSON ATTEMPTING TO ESTABLISH OWNERSHIP OF THE VOLUME MUST GIVE HIS 'USERID' TO PREVENT FUTURE UNAUTHORIZED USE OF THE VOLUME BY OTHERS USING THE CYTOS SYSTEM.

PROJID — THE PROJECT TO WHICH THE USER BELONGS. THE PROJECT ALLOWS A NUMBER OF DIFFERENT USERS ACCESS TO A VOLUME WITHOUT ALLOWING UNLIMITED ACCESS.

CART_ID — THE CARTRIDGE_ID OF THE VOLUME FOR WHICH OWNERSHIP IS TO BE ESTABLISHED. THE VOLUME MUST BE MOUNTED AT THE TIME THE JOIN COMMAND IS ISSUED.

ALL — ESTABLISHES INITIAL PERMISSION STATUS ON THE VOLUME ALLOWING UNLIMITED ACCESS. IF NOT SPECIFIED, THE PERMIT COMMAND MUST BE GIVEN TO ALLOW ACCESS BY ANYONE BUT THE NEW OWNER.

EXAMPLE: ==> JOIN SMITH CHM24 2323

IN THIS CASE, THE USER SMITH, WHO IS PART OF THE PROJECT CHM24, IS ATTEMPTING TO ESTABLISH OWNERSHIP OF THE VOLUME WHOSE CARTRIDGE_ID IS 2323.

NOTES: (1) YOU MAY NOT ESTABLISH OWNERSHIP OVER A VOLUME WHICH IS ALREADY OWNED BY ANOTHER PERSON. ONLY THE CURRENT OWNER OF A VOLUME CAN CHANGE THE OWNERSHIP, AND TO DO SO, HE (THE CURRENT OWNER) MUST BE LOGGED ON TO THE SYSTEM.

(2) TO BE ABLE TO RUN CYTOS, YOU MUST BE JOINED TO AT LEAST ONE VOLUME CURRENTLY MOUNTED ON THE SYSTEM.

(3) IF YOU ARE TRYING TO USE CYTOS FOR THE FIRST TIME, YOU MUST ISSUE THE JOIN COMMAND BEFORE THE LOGON COMMAND, TO ESTABLISH INITIAL OWNERSHIP OF THE CYTOS SYSTEM VOLUME. YOU WILL NOT BE ALLOWED TO LOGON UNTIL OWNERSHIP HAS BEEN ESTABLISHED.

(4) IF YOU ARE JOINING A SATELLITE CARTRIDGE, IT MUST BE MOUNTED BEFORE THE JOIN COMMAND, AND THEN REMOUNTED AFTER THE JOIN COMMAND IN ORDER FOR IT TO BE ACCESSED IN TERMS OF ITS NEW ATTRIBUTES.

KEYBOARD

FUNCTION: RETURN TEMPORARILY TO CONVERSATIONAL MODE FROM EXECUTE MODE OR ENTER CONVERSATIONAL MODE FROM CARD-INPUT MODE.

NOTE: WHEN ENTERED IN CONVERSATIONAL MODE, THE KEYBOARD COMMAND HAS NO EFFECT. TO RETURN TO EXECUTE MODE FROM CONVERSATIONAL MODE, USE THE CONTINUE COMMAND.

```
LINKEDIT  INDSNAME  OUTDSNAME  <OPTIONS>
```

FUNCTION: INVOKE THE IBM CORE LOAD BUILDER TO PRODUCE A LOAD MODULE FROM ONE OR MORE OBJECT MODULES.

ALIAS: LKED

PARAMETERS:

INDSNAME — THE OBJECT MODULE (A MAINLINE PROGRAM) TO BE LINKEDITED. IF NO DATA SET TYPE IS GIVEN, .O IS ASSUMED.

OUTDSNAME — THE NAME OF THE LOAD MODULE PRODUCED. IF NOT GIVEN (I.E., NULL), THE COMMAND WILL CAUSE THE LOAD MODULE TO BE NAMED INDSNAME.L. IF OUTDSNAME.L (OR INDSNAME.L IF THE SECOND PARAMETER IS NULL) ALREADY EXISTS, IT MAY NOT BE REUSED UNTIL IT HAS BEEN PREVIOUSLY ERASED. THE LINKEDIT WILL NOT PROCEED.

OPTIONS — A COLLECTION OF ONE OR MORE OF THE FOLLOWING OPTIONS TO INDICATE TO THE CORE LOAD BUILDER THE TYPE OF LOAD MODULE TO BE PRODUCED.

MAP — A CORE LOAD MAP WILL BE PRODUCED AS PART OF THE OUTPUT LISTING.

ERASE — INDSNAME WILL BE ERASED IF THE CORE LOAD IS SUCCESSFULLY BUILT.

LOCAL — MUST BE SPECIFIED IF YOU WANT ONE LOAD-ON-CALL MODULE TO BE ABLE TO CALL ANOTHER LOAD-ON-CALL MODULE.

DISK_X — INDICATES THE DISK I/O ROUTINE TO BE LOADED UPON EXECUTION OF THE LOAD-MODULE. THE CHOICE IS:

DISKZ|DISK1|DISKN

ILS — THE SPECIAL INTERRUPT LEVEL SUBROUTINES ARE TO BE LOADED IN ORDER TO ALLOW THE USE OF INDEX REGISTER 3 IN AN ASSEMBLER MODULE.

FILES — RESOLVE ALL DATA SET/FILE CORRESPONDENCES AT CORE LOAD BUILDING TIME INSTEAD OF AT EXECUTION TIME.

EXAMPLES:

==> LINKEDIT CURVE CURVE

THE OBJECT PROGRAM STORED IN THE DATA SET NAMED CURVE.O IS LINKEDITED. THE RESULTING LOAD MODULE IS NAMED CURVE.L.

==> LKED REGRS TRY$2 (ERASE,MAP)

THE OBJECT MODULE STORED IN THE DATA SET NAMED REGRS IS LINKEDITED AND THE RESULTING LOAD MODULE IS PLACED IN THE DATA SET NAMED TRY$2, IF THAT DATA SET DOES NOT ALREADY EXIST. A CORE MAP IS PRINTED ON THE SYSTEM PRINTER AND, IF THE LINKEDIT IS SUCCESSFUL, THE OBJECT MODULE REGRS WILL BE ERASED.

NOTES:

(1) THIS COMMAND ONLY PRODUCES A LOAD MODULE WHICH MAY BE EXECUTED OR RUN; IT DOES NOT PUT THE PROGRAM INTO EXECUTION. TO EXECUTE THE PROGRAM, YOU MUST ISSUE THE RUN COMMAND, OR JUST USE THE SIMPLE DATA SET NAME AS THE COMMAND 'VERB' (E.G. CURVE, OR REGRS).

(2) ANY SUBROUTINES TO BE INCLUDED VIA THE OVERLAY, INCLUDE, SUBSTITUTE, OR GRAPHICS COMMANDS MUST BE DEFINED PRIOR TO ISSUING THE CORRESPONDING LINKEDIT COMMAND. FURTHERMORE, SINCE THE OBJECT MODULE WILL RESIDE IN SYSTEM WORKING STORAGE AT THE TIME THE LOAD MODULE IS CREATED, YOU SHOULD GIVE A NULL PARAMETER () AS THE PROGRAM_NAME WITH WHICH EACH SUBROUTINE LIST IS TO BE ASSOCIATED.

(3) IF THE FILES OPTION IS GIVEN, ALL PREVIOUS NAMEFILES MUST REFERENCE DATASETS IN THE FIXED AREA OR SCRATCH DATA SETS IN WORKING STORAGE.

(4) THE OPTIONS DISKZ, DISK1, AND DISKN ARE NOT VALID UNDER CYTOS/II. THE APPROPRIATE INTERFACE TO DISKZ WILL BE LOADED AUTOMATICALLY.

LIST <DECK_COUNT> <LIMITS>

FUNCITON: PRINT ONE OR MORE CARD SOURCE FILES.

PARAMETERS:

DECK_COUNT THE NUMBER OF SOURCE CARD FILES TO BE PRINTED FROM CARDS. EACH CARD FILE MUST BE TERMINATED WITH A CONTROL CARD (ANY CARD WITH // IN COLUMNS ONE AND TWO). IF A COUNT IS OMITTED, ONE DECK WILL BE LISTED.

LIMITS A COLLECTION OF THREE DECIMAL NUMBERS INDICATING THE EXTENT OF PRINTING TO BE PERFORMED.

THE FIRST NUMBER INDICATES THE NUMBER OF CHARACTERS, STARTING WITH THE CHARACTER IN COLUMN ONE, THAT ARE TO BE PRINTED FROM EACH CARD. THE DEFAULT IS 80, THE ENTIRE CARD.

THE SECOND NUMBER INDICATES THE NUMBER OF THE FIRST RECORD OF THE CARD FILE TO BE PRINTED. THE DEFAULT IS ONE, THE FIRST CARD IN EACH FILE.

THE THIRD NUMBER INDICATES THE NUMBER OF THE INDEX OF THE LAST RECORD OF THE FILE TO BE PRINTED. THE DEFAULT IS THE LAST RECORD IN THE CARD FILE.

EXAMPLE: ==> LIST 5 (,40,60)

WOULD CAUSE FIVE CARD DECKS, SEPARATED FROM ONE ANOTHER BY ONE '//' CONTROL CARD, TO BE LISTED ON THE SYSTEM PRINTER. ONLY CARDS 40 THROUGH 60 OF EACH DECK WOULD BE PRINTED. BECAUSE THE FIRST ITEM OF THE COLLECTION IS NULL, ALL EIGHTY COLUMNS OF EACH CARD WOULD BE PRINTED. THE LIMITS LIST IS THUS EQUIVALENT TO (80,40,60).

NOTE: IF NO PARAMETERS ARE GIVEN, ONLY THE FIRST SOURCE DECK ENOCUNTERED WILL BE LISTED. FURTHERMORE, THE ENTIRE FILE WILL BE LISTED.

LOADGO DSNAME <OPTIONS>

FUNCTION: LOAD OBJECT MODULE(S) AND PLACE THE RESULTING PROGRAM IN EXECUTION; NO LOAD MODULE IS SAVED.

ALIAS: L

PARAMETERS:

DSNAME THE OBJECT MODULE CONTAINING THE MAINLINE PROGRAM TO BE LOADED AND EXECUTED, OR EXECUTE THE PROGRAM LEFT IN SYSTEM WORKING STORAGE IF NO DSNAME, ().

OPTIONS: ALL OPTIONS VALID FOR THE LINKEDIT COMMAND EXCEPT THE ERASE OPTION ARE EQUALLY VALID FOR THE LOADGO COMMAND. IF MORE THAN ONE OPTION IS ENTERED, THE COLLECTION MUST BE ENCLOSED BY PARENTHESES. THE VALID OPTIONS ARE:

MAP|LOCAL|ILS|DISKZ|DISK1|DISKN|FILES

EXAMPLES: ==> LOADGO DCOPY

THE OBJECT MODULE DCOPY.O IS LOADED, AND THE RESULTING PROGRAM IS PLACED IN EXECUTION.

==> LOADGO CARSN.PROG4.O(1003) (MAP,DISKN)

CARSN'S PROGRAM NAMED PROG4, IF IT CAN BE FOUND ON HIS CARTRIDGE WITH A VOLUME ID OF 1003, WILL BE LOADED. DISKN WILL BE USED FOR INPUT/OUTPUT OPERATIONS TO THE DISK. AFTER A CORE MAP IS PRINTED, THE PROGRAM WILL BE PLACED IN EXECUTION.

NOTES: (1) ANY SUBROUTINES TO BE INCLUDED VIA THE OVERLAY, INCLUDE, GRAPHICS, OR SUBSTITUTE COMMANDS MUST BE DEFINED PRIOR TO ISSUING THE CORRESPONDING LOADGO COMMAND. THE 'PROGRAM_NAME' PARAMETER TO THESE COMMANDS WILL BE THE SAME NAME USED FOR THE 'DSNAME' PARAMETER TO THE LOADGO COMMAND.

(2) THE OPTIONS DISKZ, DISK1, AND DISKN ARE NOT VALID UNDER CYTOS/II. THE APPROPRIATE INTERFACE TO DISKZ WILL BE LOADED AUTOMATICALLY.

LOGOFF <HOLD|SHUTDOWN>

FUNCTION: TERMINATE THE CURRENT CYTOS SESSION.

ALIAS: LOGOUT

OPTION: EITHER OF THE FOLLOWING:

HOLD — IF HOLD IS TYPED AS A PARAMETER TO THE LOGOFF COMMAND, CYTOS WILL BE READY TO ACCEPT ANOTHER LOGON COMMAND FROM ANOTHER USER. OTHERWISE, CONTROL WILL BE RETURNED TO THE DISK MONITOR SYSTEM IN BATCH MODE; FURTHER BATCH PROCESSING WILL COMMENCE WITH THE FIRST '// JOB' CARD ENCOUNTERED IN THE JOB STREAM.

SHUTDOWN — THE SHUTDOWN OPTION MAY BE SPECIFIED WITH THE LOGOFF COMMAND WHENEVER YOU WISH TO POWER DOWN THE 1130 SYSTEM. WHEN THE COMPUTER IS POWERED UP AT A LATER TIME, AND ALL PREVIOUSLY AVAILABLE DISK CARTRIDGES ARE IN THE FILE READY STATUS, YOU NEED ONLY PRESS PROGRAM START IN ORDER TO RETURN TO CYTOS COMMAND LEVEL. Y OU MAY THEN LOGON AND PROCEED AS DESIRED.

NOTES: (1) THE SHUTDOWN OPTION IS NOT VALID ON CYTOS/II SYSTEMS.

(2) DISK CARTRIDGES SHOULD NOT BE MOVED BETWEEN THE TIME THE SYSTEM IS POWERED DOWN AND LATER POWERED UP. THE SHUTDOWN OPTION SHOULD NOT BE USED IF OTHER USERS MAY HAVE ACCESS TO THE MACHINE DURING YOUR ABSENCE, AS THEY MAY INADVERTENTLY FAIL TO RESTORE ALL CARTRIDGES TO THEIR PREVIOUS DRIVES.

LOGON USERID PROJID <CARTRIDGE_ID> <LIST>

FUNCTION: INITIATE A CYTOS SESSION BY PROVIDING ACCESS TO THE CYTOS COMMAND LANGUAGE.

ALIAS: LOGIN

PARAMETERS:

USERID — THE PERSON LOGGING ON MUST IDENTIFY HIMSELF WITH HIS 'USERID' SO THAT CYTOS MAY TEST WHETHER HE IS ALLOWED ACCESS TO THE CYTOS SYSTEM AND TO THE VOLUMES CURRENTLY MOUNTED.

PROJID — A PROJECT KEY MUST ALSO BE ENTERED. TO TEST WHETHER THE USER IS A MEMBER OF THE GROUP IDENTIFIED BY 'PROJID' THAT IS ALLOWED ACCESS TO THE SYSTEM.

CART_ID — IF GIVEN, THE THIRD PARAMETER INDICATES THE VOLUME FROM WHICH THE PERSON LOGGING ON WISHES TO RUN THE CYTOS SYSTEM. THIS PARAMETER HAS MEANING ONLY WHEN ENTERED AFTER A PREVIOUS LOGOFF HOLD BY ANOTHER USER WHO HAS BEEN RUNNING CYTOS FROM A DIFFERENT CARTRIDGE.

IF THE PERSON LOGGING ON WISHES TO RE-ARRANGE THE LOGICAL ORDER OF THE VOLUMES TO BE MOUNTED ON THE VARIOUS PHYSICAL DISK DRIVES, HE MAY ENTER A COLLECTION OF CARTRIDGE_IDS, SIMILAR TO THE CORRESPONDING PARAMETER OF THE MOUNT COMMAND.

LIST — A TABLE OF ALL VOLUMES PHYSICALLY MOUNTED BUT NOT LOGICALLY AVAILABLE WILL BE PRINTED (IN ADDITION TO THE NORMAL TABLE OF VOLUME/DRIVE CORRESPONDENCES.)

NOTES: (1) IF BOTH 'USERID' AND 'PROJID' CORRESPOND TO THE OWNERSHIP OF ANY SPECIFIED VOLUME, THE USER WILL BE PROMPTED TO ENTER A PASSWORD. IF THE PASSWORD IS ALSO VALID, HE MAY CONTINUE TO ISSUE CYTOS COMMANDS. IF NOT, HE WILL BE PROMPTED AGAIN FOR ANOTHER LOGON COMMAND.

(2) CONTROL MAY BE RETURNED TO THE DISK MONITOR SYSTEM VIA THE LOGOFF COMMAND IF A USER IS UNABLE TO LOGON BY VIRTUE OF NOT KNOWING EITHER THE USERID OR CORRESPONDING PASSWORD.

(3) UNDER CYTOS/II, THE CARTRIDGE_ID OF THE MASTER CARTRIDGE MUST NOT BE INCLUDED IN THE CARTRIDGE_ID LIST.

MACLIB <HERE>

FUNCTION: INVOKE THE IBM MACRO UPDATE PROGRAM IN ORDER TO BUILD OR MODIFY A USER MACRO LIBRARY.

OPTION: IF HERE IS SPECIFIED, INPUT AND OUTPUT WILL BE AT THE TERMINAL. OTHERWISE, MACRO UPDATE COMMANDS WILL BE READ FROM CARDS AND PRINTED OUTPUT ROUTED TO THE SYSTEM PRINT DEVICE.

REFERENCE: FOR AN EXPLANATION OF THE FACILITIES OF THE MACRO UPDATE PROGRAM, SEE 'IBM 1130/1800 ASSEMBLER LANGUAGE,' SECTION 11, 'HOW DO YOU MAINTAIN MACRO LIBRARIES?', GC26-3778.

NOTES: (1) WHEN UPDATE COMMANDS ARE ENTERED FROM THE TERMINAL (HERE), THE COMMAND VERB MUST BE TYPED BEFORE ANY BLANKS BECAUSE, UNLIKE THE CYTOS COMMAND LANGUAGE, UPDATE COMMANDS ARE EXPECTED TO BE FOUND IN COLUMN 1 OF THE INPUT RECORD.

(2) IF THE HERE OPTION IS USED, AND THE MACRO LIBRARY IS LARGE, IT IS RECOMMENDED THAT THE SELECT P MACRO UPDATE COMMAND NOT BE ISSUED, TO MINIMIZE PRINTED OUTPUT.

(3) UNLIKE TYPED INPUT TO THE ASSEMBLER, THE MACRO UPDATE PROGRAM EXPECTS THE FULL SOURCE RECORD TO BE ENTERED, EVEN FROM THE TYPEWRITER. THEREFORE, TWENTY BLANKS MUST PRECEDE THE LABEL FIELD OF EACH ASSEMBLER INSTRUCTION ENTERED FROM THE TERMINAL.

MOUNT CARTRIDGE_ID_LIST <LIST>

FUNCTION: TO SET OR RESET THE LOGICAL ORDER OF DIRECT ACCESS VOLUMES TO BE ACCESSIBLE TO THE CYTOS SYSTEM.

ALIAS: M

PARAMETERS:

CART_LIST A LIST OF CARTRIDGE_IDS, IN LOGICAL ORDER, TO BE MOUNTED ON DIRECT ACCESS STORAGE DRIVES. THE FIRST ID, FOR LOGICAL DRIVE ZERO, SPECIFIES THE CARTRIDGE FROM WHICH CYTOS WILL BE RUN AFTER THE CURRENT COMMAND. CYTOS MUST RESIDE ON THIS CARTRIDGE IN ORDER FOR THE MOUNT COMMAND TO PROCEED.

LIST — A TABLE OF ALL VOLUMES PHYSICALLY MOUNTED BUT NOT LOGICALLY AVAILABLE WILL BE PRINTED (IN addition TO THE NORMAL TABLE OF VOLUME/DRIVE CORRESPONDENCES.)

EXAMPLE:

```
==> MOUNT (1111,2222,3333)
```

VOLUMES 1111, 2222, AND 3333 ARE MOUNTED ON LOGICAL DRIVES 0, 1, AND 2 RESPECTIVELY. THE CYTOS SYSTEM MUST RESIDE ON VOLUME 1111, AS THIS HAS BEEN DEFINED AS THE SYSTEM CARTRIDGE.

NOTES:

(1) THE VOLUMES WILL BE MOUNTED IN THE MANNER SPECIFIED UNTIL THE NEXT MOUNT COMMAND IS ISSUED, OR UNTIL A '// JOB' CARD IS ENCOUNTERED IN A BATCH JOB STREAM PROCESSED AS THE RESULT OF ISSUING THE CYTOS BATCH COMMAND. TO DETERMINE WHICH VOLUMES CORRESPOND TO WHICH LOGICAL DRIVES AT ANY GIVEN TIME, YOU CAN ISSUE THE CYTOS DCOM COMMAND. THE DISK COMMUNICATIONS AREA ALSO SHOWS THE LOCATION AND SIZES OF THE VARIOUS SYSTEM AREAS DEFINED ON EACH VOLUME CURRENTLY MOUNTED ON THE SYSTEM.

(2) IF OWNERSHIP OF THE VOLUME SPECIFIED FOR LOGICAL DRIVE 0 DIFFERS FROM THE VOLUME ON THAT DRIVE AT THE TIME THE MOUNT COMMAND IS ISSUED, YOU WILL BE PROMPTED TO LOGON AGAIN IN ORDER TO ESTABLISH A NEW RELATIONSHIP BETWEEN YOU AND THE CYTOS SYSTEM.

(3) IF ALL VOLUMES GIVEN IN THE CARTRIDGE_LIST ARE PHYSICALLY MOUNTED AT THE TIME THE COMMAND IS ISSUED, THE SYSTEM WILL ESTABLISH THE RELATIONSHIP BETWEEN LOGICAL DRIVES AND PHYSICAL DRIVES AUTOMATICALLY. IF A REQUESTED VOLUME IS NOT MOUNTED ON A PHYSICAL DRIVE, OR IF TWO OR MORE REQUESTED VOLUMES HAVE IDENTICAL IDENTIFICATIONS, ERROR MESSAGES TO THAT EFFECT WILL BE GIVEN.

(4) UNDER CYTOS/II, THE CARTRIDGE_ID OF THE MASTER CARTRIDGE (ON LOGICAL DRIVE 0) MUST NOT BE INCLUDED IN THE CARTRIDGE_ID LIST. THE CYTOS/II EQUIVALENT OF THE ABOVE EXAMPLE WOULD BE:

```
==> MOUNT (2222,3333)
```

VOLUME 1111 IS ALREADY MOUNTED ON LOGICAL DRIVE 0 AND IS AUTOMATICALLY AVAILABLE TO ALL USERS ON THE SYSTEM.

```
MOVE (OLD_DSNAME,...) (NEW_DSNAME,...) <ERASE>
```

FUNCTION: TO MOVE AND RENAME ONE OR MORE DATA SETS.

PARAMETERS:

OLD_DSNAME THE NAMES OF THE DATA SETS TO BE MOVED: .S, .O, .L, OR .* MAY BE SPECIFIED. OLD_DSNAME MAY BE ANY VALID CYTOS DATA SET NAME.

NEW_DSNAME THE NAMES OF THE NEW DATA SETS TO BE CREATED; IF A TYPE IS NOT GIVEN, IT DEFAULTS TO THE TYPE OF THE CORRESPONDING NAME IN THE LIST OF OLD_DSNAMES. IF NEW_DSNAME ALREADY EXISTS, NO COPY WILL TAKE PLACE UNLESS IT IS AT LEAST AS LARGE AS THE CORRESPONDING OLD_DSNAME.

OPTION: IF ERASE IS SPECIFIED AS THE THIRD PARAMETER, THE OLD DATA SETS WILL BE ERASED AFTER THE COPY HAS BEEN MADE.

EXAMPLE:

```
== MOVE (PROG.*(3333), DATA(3333)) &
   (PROGN.*(4444), DATA(4444)) ERASE
```

MOVES THE FOUR DATA SETS PROGN.S, PROGN.O, PROGN.L, AND DATA.S FROM ONE VOLUME TO ANOTHER. IN THIS EXAMPLE, BOTH VOLUMES ARE OWNED BY THE SAME PERSON AND BOTH ARE MOUNTED AT THE TIME THE MOVE COMMAND IS ISSUED.

NOTES: (1) THE DATA SET TYPES OF CORRESPONDING ITEMS IN THE LIS TS MUST ALWAYS AGREE; OTHERWISE, ALL MOVING WILL BE TERMINATED AND CYTOS WILL PROMPT FOR THE NEXT COMMAND.

(2) YOU MAY MOVE FROM AND INTO DATA SETS BELONGING TO OTHER USERS SO LONG AS THEY HAVE GIVEN YOU PERMISSION TO READ FROM ANY 'OLD_DSNAME', TO WRITE INTO ANY 'NEW_DSNAME', AND TO COPY USING THE COPY OR MOVE COMMANDS.

```
NAMEFILE FILE_PSEUDONYM  DSNAME
      <ALLOCATION_UNIT>  <SIZE>
```

FUNCTION: ESTABLISH CORRESPONDENCE BETWEEN A FILE_PSEUDONYM USED IN A PROGRAM AND THE ACTUAL DATA SET TO BE READ OR WRITTEN. THIS CORRESPONDENCE REMAINS IN EFFECT UNTIL YOU REVOKE IT BY ISSUING A NEW NAMEFILE COMMAND FOR THE SAME PSEUDONYM.

ALIASES: NF, FILES

PARAMETERS:

FILE_PSDNYM THE DECIMAL FILE NUMBER USED TO IDENTIFY THE FILE IN THE PROGRAM (E.G. THE FORTRAN DEFINE FILE IDENTIFICATION).

DSNAME ANY VALID SOURCE DATA SET EXCEPT .T, .C, AND .D; IF THE DATA SET BELONGS TO ANOTHER USER, HE MUST HAVE GIVEN YOU BOTH READ AND WRITE PERMISSION. IF THE DATA SET IS NOT FOUND, A NEW DATA SET WILL BE ALLOCATED AUTOMATICALLY, THE SIZE BEING DETERMINED BY THE LAST TWO PARAMETERS. BY SPECIFYING # YOU MAY WRITE AND THEN READ FROM A SCRATCH FILE WHICH WILL BE DISCARDED AFTER EACH EXECUTION OF THE PROGRAM.

ALLOC_UNIT THE TYPE OF UNITS TO BE ALLOCATED IF THE DATA SET IS NEW:

BLOCKS|CARDS|SECTORS|TRACKS|CYLINDERS

SIZE THE NUMBER OF UNITS TO BE ALLOCATED. IF OMITTED, THIS DEFAULTS TO 80. THIS IF BOTH THE THIRD AND FOURTH PARAMETERS ARE OMITTED AND A NEW DATA SET MUST BE CREATED, ITS SIZE WILL BE 40 CARDS.

EXAMPLE: ==> NAMEFILE 3 DATAX(2) SECTORS 30

WOULD CREATE A DATA SET DATAX WITH A SIZE OF 30 SECTORS ON THE VOLUME CURRENTLY MOUNTED ON LOGICAL DRIVE 2, AND WOULD CAUSE ANY REFERENCE TO FILE 3 BY A USER'S PROGRAM TO REFER TO THE NEW DATA SET.

NOTES: (1) ALL CORRESPONDENCES BETWEEN FILE_PSEUDONYMS AND DATA SETS NOT MADE USING THE NAMEFILE COMMAND MUST BE MADE AT EXECUTION TIME AS A RESPONSE TO THE WHATFILE QUERY (SEE SECTION 2.4, ABOVE). ANY PARAMETER VALID FOR THE NAMEFILE COMMAND (EXCEPT THE PSEUDONYM ITSELF) IS A VALID PARAMETER FOR A RESPONSE TO WHATFILE.

(2) IF YOU WISH TO DETERMINE WHAT DATA SETS ARE CURRENTLY REFERRED TO, ISSUE THE NAMEFILE COMMAND WITHOUT ANY PARAMETERS. OR, IF YOU ARE ESTABLISHING A CORRESPONDENCE, INCLUDE THE LIST OPTION IN THE THIRD PARAMETER:

```
==> NF 3 DATAX (SECTORS,LIST) 30
```

(3) TO DISASSOCIATE A DATA SET FROM A FILE_PSEUDONYM, ISSUE THE NAMEFILE COMMAND WITH THE FILE NUMBER, BUT OMIT THE DSNAME PARAMETER:

```
==> NF 6
```

(4) TO REMOVE ALL CORRESPONDENCES BETWEEN FILE_PSEUDONYMS AND DATA SETS PREVIOUSLY ESTABLISHED IN THIS SESSION, ISSUE THE CLEAR FILES COMMAND.

NEWPASS

FUNCTION: CHANGE A USER'S PASSWORD.

INPUT: THE SYSTEM PROMPTS FOR THE NEW PASSWORD: THE PASSWORD IS NOT PRINTED BACK AS IT IS ENTERED.

NOTE: THE PASSWORD IS INITIALLY SET WHEN A USER JOINS HIMSELF TO A VOLUME, BEFORE HE IS ALLOWED ACCESS TO THE SYSTEM.

NEWSPACE (DSNAME, ...) <ALLOC_UNIT> <SIZE>

FUNCTION: RESERVE DISK STORAGE SPACE FOR ONE OR MORE SOURCE DATA SETS.

ALIAS: N

PARAMETERS:

DSNAMES IF A COLLECTION OF NAMES IS GIVEN, IDENTICAL DATA SETS WILL BE CREATED.

ALLOC_UNIT THE SIZE OF THE NEW DATA SETS IS SPECIFIED IN TERMS OF ONE OF THE FOLLOWING UNITS:

BLOCKS|CARDS|SECTORS|TRACKS|CYLINDERS

SIZE THE NUMBER OF UNITS TO BE ALLOCATED FOR EACH DATA SET. IF OMITTED, 80 UNITS OF THE SPECIFIED TYPE WILL BE ALLOCATED.

EXAMPLE: ==> NEWSPACE (CRV1,CRV2,CRV3)

THREE SOURCE DATA SETS, NAMED CRV1, CRV2, AND CRV3 WILL BE CREATED. SINCE NO SPECIFIC SIZE IS GIVEN, THE DEFAULT SIZE (80 BLOCKS) WILL BE USED FOR EACH.

NOTES: (1) IF A DATA SET NAME IS MODIFIED WITH ANOTHER USER'S ID, PERMISSION TO WRITE ON HIS VOLUME MUST PREVIOUSLY HAVE BEEN GRANTED FOR THE DATA SET TO BE CREATED.

(2) IF NO SPECIFIC VOLUME OR DRIVE IS MENTIONED, THE DATA SET WILL BE CREATED ON THE VOLUME BELONGING TO THE IMPLIED USER RESIDING ON THE LOWEST NUMBERED LOGICAL UNIT. THUS, ASSUMING THE FOLLOWING VOLUMES WERE MOUNTED:

```
DRIVE 0 VOL=1111 USER=JONES
DRIVE 1 VOL=2222 USER=JONES
DRIVE 2 VOL=3001 USER=DREW
DRIVE 3 VOL=4444 USER=JONES
DRIVE 4 VOL=3002 USER=DREW
```

AND THE FOLLOWING COMMAND ISSUED BY JONES (WHO HAS PERMISSION TO WRITE ON DREW'S VOLUMES):

==> NEWSPACE (XDATA, DREW.FDATA.S) TRACKS100

THE TWO DATA SETS WOULD BE PLACED ON VOLUMES 1111 AND 3001, RESPECTIVELY, SINCE THESE VOLUMES ARE ON THE LOWEST NUMBERED LOGICAL DRIVES FOR EACH OF THE IMPLIED OR SPECIFIED USERS.

(3) THE MAXIMUM SIZE OF A FILE TO BE CREATED IS THE SIZE OF SYSTEM WORKING STORAGE AT THE TIME THE NEWSPACE COMMAND IS ISSUED.

(4) IF THE ROUND OPTION HAS PREVIOUSLY BEEN SET, THE SOURCE FILE CREATED WILL BE PADDED WITH SUFFICIENT RECORDS TO FORCE THE SIZE OF THE FILE TO AN INTEGRAL SECTOR.

```
OVERLAY  PROGRAM_NAME <(SUBROUTINE_NAMES)>  <OPTIONS>
```

FUNCTION: TO CREATE AN OVERLAY STRUCTURE FOR A PROGRAM TO BE BUILT INTO A CORE LOAD BY EITHER THE LINKEDIT OR LOADGO COMMANDS.

ALIASES: O, LOCAL

PARAMETERS:

PROGRAM — THE NAME OF THE MAINLINE PROGRAM (IN OBJECT MODULE FORM) FOR WHICH AN OVERLAY STRUCTURE IS BEING CREATED.

SUBR_NAMES — ONE OR MORE SUBROUTINES, IN OBJECT MODULE FORM (THE TYPE .O NEED NOT BE GIVEN), TO BE PLACED IN THE OVERLAY AREA AND, DURING EXECUTION OF THE PROGRAM, TO BE AUTOMATICALLY LOADED WHEN CALLED BY THE MAINLINE PROGRAM, OR BY ANOTHER SUBROUTINE. EACH SUBROUTINE MSUT BE YOURS; THAT IS, THE SIMPLE DSNAME MAY NOT BE MODIFIED WITH ANOTHER USER'S ID. IF YOU WISH TO USE SOMEONE ELSE'S SUBROUTINES, YOU MUST FIRST COPY THEM INTO YOUR OWN LIBRARY, USING THE COPY COMMAND, AND FOR THIS YOU MUST HAVE BEEN PREVIOUSLY GRANTED PERMISSION TO COPY BY THE OWNER.

OPTIONS: YOU MAY SPECIFY EITHER OF THE FOLLOWING:

APPEND — IF YOU WISH TO ADD THIS GROUP OF SUBROUTINE NAMES TO THOSE ALREADY ASSOCIATED WITH THE PROGRAM_NAME SPECIFIED. IF APPEND IS NOT GIVEN, ALL PREVIOUS OVERLAYS GIVEN FOR THE SPECIFIED MAINLINE WILL BE IGNORED.

LIST — IF YOU WISH A LIST OF ALL SUBROUTINES CURRENTLY DEFINED TO BE PARTS OF AN OVERLAY STRUCTURE.

EXAMPLE: ==> OVERLAY FRMAT (CARDZ,TYPEZ,PAPTZ) APPEND

WOULD CAUSE THE SUBROUTINES CARDZ, TYPEZ, AND PAPTZ TO BE ADDED TO THE LIST OF SUBROUTINES ALREADY TO BE INCLUDED IN THE OVERLAY AREA WHEN A LOAD MODULE IS BUILT FROM THE MAINLINE PROGRAM FRMAT BY THE CORE LOAD BUILDER.

NOTES: (1) IF A LOAD MODULE IS TO BE BUILT USING THE LINKEDIT COMMAND, THE PROGRAM_NAME PARAMETER TO THE OVERLAY COMMAND MUST BE NULL, (), BECAUSE AT THE TIME THE CORE LOAD IS BEING BUILT, THE MAINLINE WILL RESIDE IN SYSTEM WORKING STORAGE, AND HENCE WILL HAVE NO PROGRAM_NAME ASSOCIATED WITH IT.

OTHER — USERS OTHER THAN THE AUTHOR AND MEMBERS OF THE PROJECT UNDER WHICH THE DATA SETS ON THE VOLUME WERE CREATED.

ALL — ANYBODY (AUTHOR+PROJECT+OTHER).

NOTE: (1) ISSUING THE PERMIT COMMAND ALWAYS ADDS TO THE PERMISSION STATUS OF THE VOLUME. TO REVOKE PERMISSION, USE THE UNPERMIT COMMAND.

(2) ONLY THE OWNER OF A VOLUME CAN CHANGE THE PERMISSION STATUS OF THAT VOLUME.

PRINT (DSNAME, ...) <OPTIONS> <LIMITS>

FUNCTION: PRINT ONE OR MORE PACKED EBCDIC SOURCE DATA SETS.

ALIAS: P

PARAMETERS:

DSNAME — THE NAME(S) OF THE DATA SET(S) TO BE PRINTED. ONLY SOURCE DATA SETS MAY BE PRINTED, ALTHOUGH THE .S TYPE NEED NOT BE SPECIFIED, AS IT IS ASSUMED. ALL DATA SETS ARE ASSUMED TO CONTAIN RECORDS OF EIGHTY (80) EBCDIC CHARACTERS.

OPTIONS: THE FOLLOWING OPTIONS MAY BE SPECIFIED:

HERE — IF YOU WANT THE DATA SET PRINTED AT THE TERMINAL INSTEAD OF ON THE PRINCIPAL PRINT DEVICE.

ERASE — IF YOU WISH THE DATA SET TO BE ERASED FROM THE DISK AFTER IT HAS BEEN PRINTED.

LIMITS — IF YOU WISH TO PRINT ONLY A PORTION OF EACH DATA SET, YOU MAY ENTER A COLLECTION OF THREE LIMITS, AS FOLLOWS:

THE FIRST NUMBER INDICATES THE NUMBER OF CHARACTERS, STARTING WITH THE CHARACTER IN COLUMN ONE OF EACH RECORD, THAT ARE TO BE PRINTED FROM EACH RECORD. THE DEFAULT IS 80, THE ENTIRE RECORD.

THE SECOND NUMBER INDICATES THE INDEX OF THE FIRST RECORD OF THE DATA SET TO BE PRINTED. THE DEFAULT IS ONE, THE FIRST RECORD OF THE DATA SET.

(2) IF YOU WISH TO DISASSOCIATE A MAINLINE NAME FROM A SET OF SUBROUTINES PREVIOUSLY ASSOCIATED, ISSUE THE OVERLAY COMMAND WITH A NULL LIST OF SUBROUTINES:

==> OVERLAY FRMAT ()

IF YOU WISH TO REMOVE ALL PREVIOUSLY DEFINED OVERLAY STRUCTURES FROM THE SYSTEM CONTROL RECORD AREA, ISSUE THE CLEAR OVERLAYS COMMAND.

(3) IF YOU WISH TO DETERMINE WHICH SUBROUTINES ARE CURRENTLY ASSOCIATED WITH WHICH MAINLINE PROGRAMS, ISSUE THE OVERLAY COMMAND WITHOUT ANY PARAMETERS.

PERMIT CARTRIDGE_ID ACCESS PERMITTEE_CLASS

FUNCTION: ALLOW OTHER USERS TO ACCESS DATA SETS ON A VOLUME.

OUTPUT: UPDATED PERMISSION STATUS OF THE VOLUME.

EXAMPLE:

```
==> PERMIT 3023 (READ,WRITE)PROJECT
AUTHOR   R W C E
PROJECT  R W
OTHER
***END OF PERMIT LIST***
```

PARAMETERS:

CART_ID — THE CARTRIDGE_ID OF THE VOLUME TO WHICH ACCESS IS BEING GRANTED. ONLY ONE VOLUME MAY BE SPECIFIED PER COMMAND.

ACCESS

- READ — PERMITTEE MAY READ DATA SETS.
- WRITE — PERMITTEE MAY WRITE INTO DATA SETS.
- COPY — PERMITTEE MAY COPY DATA SETS USING THE COPY COMMAND.
- ERASE — PERMITTEE MAY ERASE DATA SETS, USING THE ERASE COMMAND, OR ERASE OPTION TO A COMMAND.

PERMITTEE

- AUTHOR — THE PERSON WHO CREATED THE DATA SETS ON THE SPECIFIED VOLUME (I.E., THE OWNER OF THE VOLUME).
- PROJECT — MEMBERS OF THE PROJECT UNDER WHICH THE DATA SETS WERE CREATED, OTHER THAN THE AUTHOR.

THE THIRD NUMBER INDICATES THE NUMBER OF THE INDEX OF THE LAST RECORD OF THE DATA SET TO BE PRINTED. THE DEFAULT IS THE LAST RECORD OF THE DATA SET.

EXAMPLES: ==> PRINT (PROGA,PROGB,PROGC.S) HERE

WILL CAUSE THE THREE SOURCE DATA SETS NAMED PROGA, PROGB, AND PROGC TO BE PRINTED AT THE TERMINAL. EACH DATA SET WILL BE PRINTED IN ITS ENTIRETY BECAUSE NO LIMITS WERE SPECIFIED.

==> P MRDNG () (20,101,199)

WOULD CAUSE THE FIRST TWENTY CHARACTERS OF RECORDS 101 THROUGH AND INCLUDING 199 OF THE SOURCE DATA SET NAMED MRDNG TO BE PRINTED ON THE PRINCIPAL PRINT DEVICE.

==> PRINT DIXON.FORTX.S ERASE

PRINT DIXON'S SOURCE PROGRAM FORTX ON THE PRINCIPAL PRINTER, THEN ERASE THE DATA SET FROM HIS VOLUME. DIXON MUST HAVE GIVEN PERMISSION TO READ IN ORDER FOR THE DATA SET TO BE PRINTED, AND PERMISSION TO ERASE IN ORDER FOR THE ERASE OPTION TO TAKE EFFECT.

PUNCH (DSNAME, ...) <ERASE>

FUNCTION: PUNCH ONE OR MORE PACKED EBCDIC SOURCE FILES ON THE SYSTEM OUTPUT DEVICE.

PARAMETERS:

DSNAMES THE NAME(S) OF THE DATA SET(S) TO BE PUNCHED. ALL VALID EBCDIC CHARACTERS WILL BE PUNCHED, ONE CHARACTER PER COLUMN. DATA SETS TO BE PUNCHED ARE ASSUMED TO CONTAIN EIGHTY (80) EBCDIC CHARACTERS PER RECORD.

ERASE MAY BE SPECIFIED IF YOU WISH THE SOURCE FILE TO BE ERASED FROM THE DISK AFTER IT HAS BEEN PUNCHED.

EXAMPLE: ==> PUNCH (FDATA,PDATA) ERASE

THE DATA SETS FDATA AND PDATA WILL BE PUNCHED TO CARDS (OR TO PAPERTAPE, IF THAT IS THE PRINCIPAL OUTPUT DEVICE); THE DATA SETS WILL BE ERASED IMMEDIATELY AFTER BEING PUNCHED.

READ (DSNAME, ...) <HERE>

FUNCTION: ACCEPT ONE OR MORE EBCDIC SOURCE FILES FROM THE SYSTEM INPUT DEVICE, AND CREATE DISK DATA SETS FROM THEM.

PARAMETERS:

DSNAME — THE NAME(S) OF THE DATA SET(S) TO BE CATALOGUED AFTER THE INFORMATION HAS BEEN READ FROM THE INPUT DEVICE. ALL DATA SETS CREATED IN THIS MANNER ARE SOURCE (TYPE .S IS ASSUMED AND NEED NOT BE ENTERED); ALL INFORMATION IS ASSUMED TO BE EIGHTY EBCDIC CHARACTERS PER INPUT RECORD.

HERE — INPUT WILL BE ENTERED FROM THE TERMINAL INSTEAD OF FROM THE PRINCIPAL INPUT DEVICE. EACH RECORD WILL BE "PADDED" WITH TRAILING BLANKS IF THE FULL 80 CHARACTERS ARE NOT TYPED BEFORE THE 'EOF' KEY IS PRESSED TO SIGNAL THE END OF EACH INPUT RECORD.

EXAMPLE: ==> READ (NEWA,NEWB)

READ INFORMATION FROM TWO CARD FILES AND CREATE TWO DATA SETS ON THE DISK, NAMING THEM NEWA AND NEWB.

NOTES:

(1) EACH SOURCE FILE MSUT BE TERMINATED BY A CONTROL RECORD (ANY RECORD WITH '//' IN THE FIRST TWO COLUMNS OF THE RECORD). READING WILL CONTINUE UNTIL A CONTROL RECORD IS ENCOUNTERED.

(2) IF THE ROUND OPTION HAS PREVIOUSLY BEEN SET, THE SOURCE FILE CREATED WILL BE ROUNDED TO THE NEAREST DISK SECTOR. THE CONTENTS OF THE PADDING AT THE END OF THE FILE WILL BE UNPREDICATABLE.

(3) UNDER CYTOS/II, YOU MUST SPECIFY HERE. THE PRINCIPAL INPUT DEVICE MAY BE USED ONLY BY THE BATCH TASK. TO READ FROM CARDS, YOU MUST SUBMIT A CARD-INPUT JOB VIA THE BATCH FACILITY.

```
RECATALOGUE  DSNAME  <NDUP>  <NN>
```

FUNCTION: REPLACE THE CONTENTS OF A CURRENTLY CATALOGUED DATA SET WITH THE DATA SET THAT IS CURRENTLY IN SYSTEM WORKING STORAGE.

PARAMETERS:

DSNAME — THE NAME OF THE DATA SET WHOSE CONTENTS WILL BE REPLACED.

NDUP — INDICATES THAT THE OLD DATA SET WILL BE REPLACED ONLY IF THE NEW DATA SET IN WORKING STORAGE CONTAINS NO ERRORS FROM A PREVIOUS COMPILATION OR ASSEMBLY. IF THIS OPTION IS NOT SPECIFIED, YOU MAY SAVE THE OBJECT MODULE PRODUCED BY A COMPILATION, EVEN THOUGH IT MAY CONTAIN ERRORS. THIS PROCEDURE IS NOT ADVISABLE UNLESS YOU ARE SURE THAT THESE ERRORS WILL HAVE NO EFFECT ON SUBSEQUENT EXECUTION OF THE PROGRAM.

NN — ANY VALID SUBROUTINE SUB-TYPE MAY BE SPECIFIED AS THE THIRD PARAMETER IF YOU WISH THE OBJECT MODULE TO BE INCLUDED IN THE SPECIAL SYSTEM OVERLAY AREA WHEN IT IS REFERENCED DURING THE BUILDING OF A CORE LOAD.

NOTES:

(1) YOU MAY NOT RECATALOGUE A DATA SET IF THE NEW VERSION IN WORKING STORAGE IS LARGER THAN THE OLD VERSION, OR

(2) IF THE TYPE OF DATA SET IN WORKING STORAGE DIFFERS FROM THE OLD DATA SET PREVIOUSLY CATALOGUED, OR

(3) IF THE OBJECT MODULE IS A SUBROUTINE AND THE ENTRY POINT NAME DIFFERS FROM THE DATA SET NAME.

REFORMAT CARTRIDGE_ID

FUNCTION: TO REFORMAT THE CONTENTS OF 'SYSTEM WORKING STORAGE' ON A SPECIFIED DISK VOLUME, AND RE-WRITE THE SECTOR ADDRESS AT THE BEGINNING OF EACH SECTOR.

PARAMETER: THE ONLY VALID PARAMETER IS THE CARTRIDGE_ID OF THE VOLUME WHOSE WORKING STORAGE AREA IS TO BE REFORMATTED. THE VOLUME MUST BE MOUNTED AT THE TIME THE COMMAND IS ISSUED. IF THIS PARAMETER IS OMITTED, SYSTEM WORKING STORAGE AS DEFINED IN THE PREVIOUS ASSIGN COMMAND OR IN THE IPL PROCEDURE WILL BE REFORMATTED.

NOTE: THE CONTENTS OF THE WORKING STORAGE PRIOR TO ISSUING THE REFORMAT COMMAND ARE DESTROYED. IF YOU HAVE A FILE IN WORKING STORAGE, THEREFORE, YOU SHOULD CATALOGUE IT IN YOUR LIBRARY BEFORE ISSUING THE REFORMAT COMMAND.

REMOVE (DSNAME, ...) <ERASE>

FUNCTION: TO MOVE ONE OR MORE DATA SETS TO AUXILIARY STORAGE WHEN NOT NEEDED ONLINE.

PARAMETER: A LIST OF DATA SETS TO BE REMOVED; ALL AUXILIARY FILES WILL BE PUNCHED IN BINARY FORMAT, WITH AN EIGHT-DIGIT EBCDIC SEQUENCE IDENTIFICATION IN COLUMNS 73-80 OF EACH BINARY RECORD. ALL TYPES OF DATA SETS MAY BE REMOVED (.S, .O, .L, AND .*): THE TYPES MAY BE IN ANY ORDER. ALL DATA SETS WILL BE ERASED IF SO SPECIFIED.

EXAMPLE: ==> REMOVE (P1$R2.*,GEORG.NDSET.S,MODIF.O(4041))

WOULD CAUSE THE FOLLOWING DATA SETS TO BE MOVED TO AUXILIARY STORAGE VIA THE PRINCIPAL I/O DEVICE, BUT NOT TO BE UNCATALOGUED FROM THE USER'S LIBRARY:

```
P1$R2.S
P1#R2.O
P1#R2.L
GEORG.NDSET.S
MODIF.O(4041)
```

(SUCCESSFUL REMOVAL DEPENDS UPON APPROPRIATE PERMISSION HAVING BEEN GRANTED TO READ AND ERASE THE DATA SETS SPECIFIED, AND UPON THE SPECIFIED VOLUMES BEING MOUNTED AT THE TIME THE COMMAND IS ISSUED).

NOTES: AUXILIARY FILES MAY BE MOVED BACK TO THE DISK AND AUTOMATICALLY CATALOGUED AGAIN BY ISSUING THE RESTORE COMMAND.

REMOVED FILES ARE IN BINARY FORMAT, AND ARE HENCE UNSUITABLE FOR INPUT TO THE SYSTEM VIA THE READ COMMAND OR ANY OTHER MODE THAT ASSUMES EBCDIC FORMAT.

RENAME OLD_DSNAME NEW_DSNAME

FUNCTION: CHANGE THE NAME OF A DATA SET FROM OLD_DSNAME TO NEW_DSNAME.

PARAMETERS:

OLD_DSNAME ONLY DATA SETS BELONGING TO THE PERSON LOGGED ON MAY BE RENAMED. THE DSNAME MAY BE QUALIFIED BY S, O, OR L; IF NO TYPE IS SPECIFIED, .S IS ASSUMED.

NEW_DSNAME MAY NOT BE AN EXISTING DATA SET. IF NOT QUALIFIED BY A TYPE, THAT OF OLD_DSNAME WILL BE ASSUMED.

NOTE: YOU MAY NOT SPECIFY CONTRADICTORY DATA SET TYPES (I.E., CHANGE THE TYPE OF A DATA SET USING THE RENAME COMMAND); THE TYPES OF OLD_DSNAME AND NEW_DSNAME MUST AGREE, WHETHER EXPLICITLY OR IMPLICITLY.

REPRODUCE <NN>

FUNCTION: CREATE A DUPLICATE COPY OF AN EBCDIC CARD FILE.

PARAMETER: A COUNT OF THE NUMBER OF SOURCE CARD FILES TO BE REPORDUCED. EACH EBCDIC CARD FILE MUST BE TERMINATED BY A CONTROL CARD (ANY CARD WITH '//' IN COLUMNS ONE AND TWO). IN ADDITION, IF THE SYSTEM INPUT/OUTPUT DEVICE IS A 1442 MODLE 6 OR 7, EACH SOURCE FILE MUST BE FOLLOWED BY AT LEAST AS MANY BLANK CARDS ON WHICH THE COPY IS TO BE PUNCHED. THESE BLANK CARDS MUST IN TURN BE TERMINATED BY BEGINNING OF SUBSEQUENT SOURCE INPUT FILES.

NOTE: REPORDUCE MAY NOT BE USED TO COPY BINARY CARD FILES. REFER TO SECTION 10.G, 'ON REPRODUCING BINARY DECKS', FOR AN EXPLANATION OF HOW TO COPY BINARY DECKS.

RESTART DSNAME <OPTIONS>

FUNCTION: RESTART AN INTERRUPTED CORE LOAD PREVIOUSLY SAVED BY THE CHECKPOINT COMMAND.

PARAMETERS:

DSNAME NAMES THE SOURCE DATA SET IN WHICH THE INTERRUPTED PROGRAM HAS BEEN SAVED.

OPTIONS:

RESUME EXECUTION OF THE CHECKPOINTED PROGRAM WILL BE RESUMED IMMEDIATELY UPON ISSUING THE RESTART COMMAND. IF NOT GIVEN, THE CORE LOAD IS BROUGHT INTO CORE ONLY ABOVE 4K AND THE FIRST 4K REMAINS IN THE CORE IMAGE BUFFER WHERE PLACED WHEN THE 'INT REQ' KEY WAS PRESSED. CONTROL IS RETURNED TO CYTOS COMMAND LEVEL WHERE OTHER COMMANDS MAY BE GIVEN BEFORE THE RESUME COMMAND ACTUALLY CAUSES THE RESUMPTION OF PROGRAM EXECUTION.

EDIT MUST BE SPECIFIED IF RESUME IS SPECIFIED AND IF THE INTERRUPTED PROGRAM CONSISTS OF AN EDITING SESSION PREVIOUSLY FROZEN

BY MEANS OF ISSUING THE EDITOR'S ATTN COMMAND.

EXAMPLE: ==> RESTART SAVEFILE RESUME

THE PROGRAM SAVED IN THE SOURCE DATA SET SAVEF.S WILL BE RELOADED AND IMMEDIATELY RESUMED FORM THE POINT AT WHICH IT HAD BEEN INTERRUPTED AND FROZEN BY THE CHECKPOINT COMMAND.

NOTES: (1) PLEASE REFER TO THE CHECKPOINT COMMAND DESCRIPTION NOTES FOR AN EXPLANATION OF THE CIRCUMSTANCES MAKING PROGRAM RESUMPTION IMPOSSIBLE.

(2) IF A USER COMMAND (INVOKED BY THE RUN COMMAND) HAS INCORPORATED SPECIAL HANDLING FOR LEVEL-4 INTERRUPTS, ADDITIONAL CODING MUST BE INCLUDED TO MAKE PROGRAM RESUMPTION POSSIBLE. (REQUEST A LISTING OF THE CYTOS ILSX4 INTERRUPT LEVEL SUBROUTINE.)

RESTORE (DSNAME, ...) (COUNT, ...)

FUNCTON: MOVE TO DISK FROM AUXILIARY STORAGE, THEN CATALOGUE, ONE OR MORE DATA SETS PREVIOUSLY REMOVED TO AUXILIARY STORAGE VIA THE REMOVE COMMAND.

PARAMETERS:

DSNAME THE NAME(S) OF THE DATA SET(S) TO BE RESTORED ON LINE. ANY TYPE OF REMOVED FILE MAY BE RESTORED: S, O, OR L. HOWEVER, THE AUXILIARY FILE MUST BE IN BINARY (AS OPPOSED TO EBCDIC) FORMAT. EACH DATA SET WILL BE CATALOGUED IN THE LOCATION EQUIVALENCE TABLE (LET) WITH THE NAME GIVEN IN THE DSNAME COLLECTION.

COUNT A RECORD COUNT MUST BE GIVEN FOR EACH FILE TO BE RESTORED. NOTE THAT IF .* WAS GIVEN AS THE DATA SET TYPE IN THE DSNAME COLLECTION, THREE COUNTS CORRESPONDING TO THE THREE POSSIBLE TYPES OF DATA SET WILL BE EXPECTED IN THE COUNT COLLECTION. DSNAMES AND COUNTS MUST HOLD CORRESPONDING POSITIONS IN EACH COLLECTION.

EXAMPLE: ==> RESTORE (ACCTR.*,DATAR.S) (206,33,42,240)

THE DATA SETS ACCTR.S, ACCTR.O, ACCTR.L, AND DATAR.S WILL BE RESTORED TO DISK FROM AUXILIARY STORAGE. A RECORD COUNT IS PROVIDED FOR EACH OF THE DATA SETS TO BE RESTORED.

RESUME <EDIT>

FUNCTION: RESUME EXECUTION OF A PROGRAM THAT HAS BEEN INTERRUPTED AND FROZEN.

PRESSING THE 'INT REQ' KEY DURING EXECUTION OF USER-WRITTEN COMMANDS AND SOME CYTOS COMMANDS CAUSES THE EXECUTION OF THAT PROGRAM TO BE SUSPENDED AND THE SYSTEM TO RETURN TO COMMAND LEVEL.

WHILE AT COMMAND LEVEL, ANY COMMAND MAY BE ISSUED THAT WILL NEITHER DESTROY THE CONTENTS OF THE CORE IMAGE BUFFER, NOR ALTER ANY OF THE FILE-PROTECT ADDRESSES ON ANY OF THE CARTRIDGES, NOR USE MORE THAN THE FIRST 4K OF CORE STORAGE. THE TABLE IN APPENDIX A INDICATES WHICH COMMANDS MAY BE ISSUED WITHOUT DESTROYING THE INTERRUPTED PROGRAM.

WHEN YOU WISH TO RETURN TO THE INTERRUPTED PROGRAM, ISSUE THE RESUME COMMAND AND EXECUTION WILL CONTINUE FROM THE POINT OF INTERRUPTION.

PARAMETER: EDIT MUST BE TYPED AFTER RESUME IF YOU WISH TO RESUME EDITING AFTER HAVING INTERRUPTED AN EDITING SESSION BY ISSUING THE EDITOR COMMAND ATTN; SEE THE EDITOR TUTORIAL, SECTION 5 ABOVE, FOR FURTHER DETAILS.

NOTES: (1) THE CYTOS CHECKPOINT AND RESTART COMMANDS ALLOW YOU TO USE ALL OF CORE AND DISREGARD THE LIMITATIONS ON THE USE OF SYSTEM WORK AREAS DURING THE INTERRUPTION OF AN EXECUTING PROGRAM.

(2) IF YOU ATTEMPT TO RESUME AN INTERRUPTED PROGRAM AND ITS STATUS IS NOT SUITABLE FOR RESUMPTION, YOU WILL BE ISSUED A MESSAGE TO THIS EFFECT.

```
RPG  INDSNAME  <OUTDSNAME>  <OPTIONS>
```

FUNCTION: INVOKE THE IBM REPORT PROGRAM GENERATOR AGAINST AN RPG SOURCE PROGRAM.

PARAMETERS:

INDSNAME — INDSNAME IS THE NAME OF THE RPG SOURCE PROGRAM TO BE COMPILED. YOU MAY SPECIFY .C IF THE RPG SOURCE PROGRAM IS TO BE READ DIRECTLY FROM CARDS, OR .T IF YOU HAVE THE PATIENCE TO ENTER AN RPG SOURCE PROGRAM FROM YOUR TERMINAL.

OUTDSNAME — OUTDSNAME IS THE NAME OF THE OBJECT MODULE PRODUCED AS A RESULT OF THE COMPILATION. THE TYPE NEED NOT BE GIVEN: .O IS ASSUMED. IF OUTDSNAME IS OMITTED, NO OBJECT MODULE WILL BE PRODUCED; IF A NULL PARAMETER () IS SPECIFIED, AN OBJECT MODULE NAMED INDSNAME.O WILL BE CATALOGUED. IF OUTDSNAME.O (OR INDSNAME.O, IF A NULL PARAMETER) ALREADY EXISTS, IT WILL BE REUSED, SO LONG AS THE NEW VERSION IS NO LARGER THAN THE OLD. IF YOU WISH TO SAVE THE OBJECT MODULE ON CARDS, SPECIFY .C; IF YOU WISH NO OBJECT MODULE AT ALL, BUT WISH TO ENTER FURTHER OPTIONS, SPECIFY .D (OR .DUMMY).

OPTIONS: THE FOLLOWING OPTIONS ARE VALID:

LIST	IF YOU WISH A LISTING ONLY.
COMPILE	IF YOU WANT A COMPILATION ONLY.
ALL	IF YOU WANT BOTH A LISTING AND A COMPILATION.
HERE	IF YOU WISH PRINTED OUTPUT AT YOUR TERMINAL INSTEAD OF ON THE PRINCIPAL PRINT DEVICE.

EXAMPLE:

```
==> RPG .C ACREC (HERE,ALL)
```

REQUESTS THE COMPILATION OF AN RPG SOURCE PROGRAM FROM CARDS. THE OBJECT MODULE, IF SUCCESSFULLY PRODUCED, WOULD BE CATALOGUED UNDER THE NAME ACREC.O; THE LISTING OF THE SOURCE PROGRAM WOULD BE MADE AT YOUR TERMINAL.

NOTES:

(1) THE RPG COMPILER REQUIRES 8K CORE STORAGE TO OPERATE.

(2) OPTIONS GIVEN WITH THE RPG COMMAND WILL OVERRIDE THOSE SPECIFIED ON THE 'H' CARD. THE PROGRAM NAME ON THE 'H' CARD WILL BE OVERRIDDEN BY THE OUTDSNAME GIVEN WITH THE COMMAND.

```
RUN  DSNAME  <OPTION>  'STRING'
```

FUNCTION: EXECUTE A USER-WRITTEN PROGRAM.

ALIAS: R

PARAMETERS:

DSNAME — THE DSNAME MUST BE OF TYPE .L. A PROGRAM BELONGING TO ANOTHER USER MAY BE SPECIFIED IF PERMISSION TO READ HAS BEEN GRANTED.

OPTION: EITHER OF THE FOLLOWING MAY BE SPECIFIED:

ASIS — DATA SET/FILE REFERENCE CORRESPONDENCES ARE ASSUMED TO HAVE BEEN RESOLVED VIA PREVIOUS RUN OR LINKEDIT. THIS OPTION SHOULD BE USED WITH GREAT CARE, SINCE THE LOAD MODULE MAY POINT TO THE WRONG PLACE ON THE DISK IF THE REFERENCED DATA SET HAS BEEN ERASED OR IS IN THE USER AREA FOLLOWING ANOTHER DATA SET WHOSE SIZE HAS BEEN ALTERED. THIS PROBLEM DOES NOT EXIST IN DMS BATCH MODE BECAUSE LOAD MODULES ARE NOT ALLOWED TO REFERENCE DATA SETS IN THE USER AREA.

NOGO — IF NOGO IS SPECIFIED, DEFINED FILES WILL BE LOOKED UP, BUT THE LOAD MODULE WILL SUBSEQUENTLY NOT BE PLACED INTO EXECUTION. THIS PROCEDURE MUST BE USED IF THE LOAD MODULE IS TO BE LINKED TO AT EXECUTION TIME BY ANOTHER LOAD MODULE; ALL DATA SET/FILE CORRESPONDENCES MUST BE RESOLVED PRIOR TO THE LINK BECAUSE THIS FUNCTION CANNOT BE PERFORMED BY THE CORE IMAGE LOADER AT LINK TIME.

PARM — IF PARM IS SPECIFIED, YOU CAN SPECIFY A QUOTED STRING WHICH CAN BE RETRIEVED IN THE PROGRAM BEING RUN WITH A CALL TO ONE OF THE FOLLOWING SUBROUTINES:

** CALL RUNF1 (IPARM) **

FOR FORTRAN A1 FORMAT (BACKWARDS IN CORE, 1 CHARACTER PER WORD LEFT JUSTIFIED).

** CALL RUNF2 (IPARM) **

FOR FORTRAN A2 FORMAT (BACKWARDS IN CORE, 2 CHARACTERS PER WORD).

** CALL RUNF4 (IPARM) **

FOR FORTRAN A4 FORMAT (BACKWARDS IN CORE, 4 CHARACTERS PER USASII WORD).

** CALL RUNA1 (IPARM) **

FOR COBOL/ASSEMBLER A1 FORMAT (FORWARDS IN CORE, 1 RIGHT JUSTIFIED CHARACTER PER WORD).

** CALL RUNA2 (IPARM) **

FOR COBOL/ASSEMBLER A2 FORMAT (FORWARDS IN CORE, 2 EBCDIC CHARACTERS PER WORD).

THE ABOVE SUBROUTINES DO THE FOLLOWING:

MOVES "STRING" TO "IPARM". "IPARM" MUST BE DIMENSIONED LARGE ENOUGH TO HOLD 80 CHARACTERS. ONLY THE NUMBER OF CHARACTERS SUPPLIED WILL BE MOVED INTO THE ARRAY BY THE SUBROUTINE. THE SUBROUTINE RETURNS DIRECTLY IF "PARM" IS NOT SPECIFIED OR IF "STRING" IS OMITTED.

NOTE: YOU MAY USE THE SIMPLE DSNAME OF A LOAD MODULE AS A COMMAND ITSELF; IT MAY NOT BE QUALIFIED BY EITHER A USERID OR A TYPE, HOWEVER. THIS, THE COMMAND:

==> RUN PROGX.L

IS EQUIVALENT TO THE COMMAND

==> PROGX

THE LOAD MODULE IN QUESTION MUST HAVE BEEN BUILT BY YOUR HAVING PREVIOUSLY ISSUED THE LINKEDIT COMMAND. IF THE LOAD MODULE WAS BUILT ON A MACHINE OTHER THAN THE ONE ON WHICH IT IS TO BE RUN, THE MACHINE ON WHICH IT IS TO BE RUN MUST BE AT LEAST AS BIG AS THE MACHINE ON WHICH THE LOAD MODULE WAS ORIGINALLY BUILT.

```
SECTOR  DISK_SECTOR  ADDR  VERIFY  (NEWVAL, ...)
```

FUNCTION: PATCH INFORMATION IN A SPECIFIED DISK SECTOR.

PARAMETERS:

DISK_SCTR — DISK SECTOR TO BE PATCHED (HEXADECIMAL). TO SPECIFY A LOGICAL DRIVE THE DISK_SECTOR MUST CONSIST OF 4 HEXADECIMAL DIGITS THE FIRST DIGIT BEING THE LOGICAL DRIVE CODE.

ADDR — RELATIVE WORK WITHIN THE SECTOR .

VERIFY — THE VALUE OF THE FIRST WORD TO BE CHANGED -- USED TO MAKE SURE YOU ARE AT THE CORRECT LOCATION.

NEWVALS — A LIST OF UP TO 20 HEXADECIMAL VALUES TO BE STORED IN THE LOCATIONS "ADDR" THRU "ADDR+19".

EXAMPLE: ==> SECTOR 18 120 333 (0303 0404)

WOULD CHANGE LOCATIONS /120 AND /121 TO THE VALUES IN THE REPLACEMENT COLLECTION ASSUMING THAT THE VERFICATION VALUE MATCHES WHAT IS ALREADY THERE IN DISK SECTOR /18.

TO PUT THE SAME CHANGE ON LOGICAL DRIVE ONE YOU WOULD ENTER THE FOLLOWING:

==> SECTOR 1018 120 333 (0303 0404)

MESSAGE: ZP04 OLD=XXXX, GIVEN=XXXX

INDICATES AN INVALID VERIFICATION. THE MESSAGE DISPLAYS THE OLD (EXISTING) VALUE AND THE VALUE YOU ATTEMPTED TO SUBSTITUTE.

SET OPTION ON|OFF

FUNCTION: MODIFY SESSION DEFAULT CONDITIONS.

PARAMETERS:

OPTION — ONE OF THE FOLLOWING CONDITIONS MAY BE SET OR RESET:

ROUND — IF ON, DATA SETS CREATED BY THE ALLOCATE, NEWSPACE, OR READ COMMANDS, AND BY THE EDITOR FILE COMMAND, WILL CONTAIN SUFFICIENT PADDING RECORDS TO FORCE AN INTEGRAL SECTOR BOUNDARY IN ACCORDANCE WITH DMS CONVENTIONS. THIS OPTION SHOULD BE USED IF YOU FREQUENTLY CREATE DATA SETS IN THE FIXED AREA. IF OFF, THE DEFAULT, DATA SETS WILL CONTAIN ONLY AS MANY RECORDS AS THE DATA ACTUALLY WARRANTS.

DUMP — IF ON, SYSTEM CORE DUMPS UNDER CYTOS/II WILL BE PRINTED ON THE SYSTEM PRINTER; IF OFF, DUMPS WILL BE PRINTED AT THE USER'S TERMINAL. THIS OPTION HAS NO EFFECT IN CYTOS/I: ALL CORE DUMPS ARE SENT TO THE PRINCIPAL PRINTER.

TELL — IF ON, THE DEFAULT, THE CYTOS/II USER MAY RECEIVE MESSAGES FROM OTHER USERS VIA THE TELL COMMAND. IF OFF, ONLY THE SYSTEM OPERATER MAY FORCE A MESSAGE TO BE PRINTED AT THE USER'S TERMINAL. THE OFF CONDITION SHOULD BE SET WHEN IT IS IMPORTANT THAT OUTPUT TO THE TERMINAL NOT BE RUINED BY EXTRANEOUS LINES.

FOLDING — UNLESS YOU ARE USING THE EDITOR, ALL INPUT TO CYTOS/II FROM YOUR TERMINAL WILL AUTOMATICALLY BE TRANSLATED TO UPPER CASE IF FOLDING IS ON, THE DEFAULT. WITHIN THE EDITOR, FOLDING IS CONTROLLED BY THE EDITOR'S SET COMMAND. (SEE THE EDITOR TUTORIAL, SECTION5.)

ON|OFF — THE NEW SETTING FOR THE CONDITION.

```
SUBSTITUTE (REFERENCED_DSNAME, ...)
   (SUBSTITUTE_DSNAME, ...) <OPTIONS>
```

FUNCTION: PROVIDE SUBSTITUTE SUBROUTINES TO BE CALLED INSTEAD OF THOSE EXPLICITLY REFERENCED DURING THE BUILDING OF A CORE LOAD.

ALIAS: EQUATE

PARAMETERS:

REFR_DSN A LIST OF ONE OR MORE NAMES APPEARING AS OPERANDS TO CALL INSTRUCTIONS OR LIBF INSTRUCTIONS IN OBJECT MODULES TO BE COMBINED INTO A CORE LOAD. THESE NAMES MUST NOT BE QUALIFIED WITH A USERID; ALL OBJECT MODULES REFERENCED IN BUILDING A CORE LOAD MUST ALREADY BE IN YOUR OWN LIBRARY.

SUBST_DSN A LIST EQUAL IN LENGTH TO THE PREVIOUS PARAMETER INDICATING WHICH OBJECT MODULES ARE TO BE SUBSTITUTED FOR EACH OF THE REFERENCED DSNAMES.

OPTIONS: THE FOLLOWING OPTIONS ARE VALID:

APPEND WILL ADD THE SPECIFIED SUBSTITUTIONS TO THOSE SPECIFIED IN EARLIER SUBSTITUTE COMMANDS. IF APPEND IS OMITTED, PREVIOUS SPECIFICATIONS WILL BE IGNORED. A MAXIMUM OF 25 SUBSTITUTIONS MAY BE IN EFFECT AT ANY GIVEN TIME.

LIST WILL PROVIDE A LIST OF ALL SUBROUTINE NAME SUBSTITUTIONS CURRENTLY IN EFFECT

```
SUBROUTINE SUBSTITUTIONS...
  TYPEZ FOR CARDZ
  TPAUZ FOR PAUSE
```

EXAMPLE:

```
==> SUBSTITUTE (NFADD,NFDIV) (FADD,FDIV.O) &
    (APPEND, LIST)
```

WOULD CAUSE FADD.O TO BE USED INSTEAD OF NFADD.O, AND FDIV.O TO BE USED INSTEAD OF NFDIV.O WHENEVER EITHER OF THE FORMER NAMES WERE ENCOUNTERED WHEN BUILDING A CORE LOAD. THESE SUBSTITUTIONS ARE ADDED TO THOSE ALREADY IN EFFECT, AND A LIST OF ALL CURRENT SUBROUTINE SUBSTITUTIONS IS PRINTED AT THE TERMINAL FOR VERIFICATION.

NOTE: THE SUBSTITUTE COMMAND IS MOST USEFUL WHEN YOU HAVE MULTIPLE VERSIONS OF ONE SUBROUTINE AND YOU WISH TO USE THEM IN RANDOM FASHION WITHOUT HAVING TO RECOMPILE THE CALLING PROGRAM EACH TIME A NEW VERSION IS TO BE USED.

```
TYPE (DSNAME, ...) <OPTIONS> <LIMITS>
```

FUNCTION: TYPE ONE OR MORE PACKED EBCDIC SOURCE DATA SETS. ON THE CONSOLE TYPEWRITER.

ALIAS: T

PARAMETERS:

DSNAME THE NAME(S) OF THE DATA SET(S) TO BE TYPED. ONLY SOURCE DATA SETS MAY BE TYPED, ALTHOUGH THE .S TYPE NEED NOT BE SPECIFIED, AS IT IS ASSUMED. ALL DATA SETS ARE ASSUMED TO CONTAIN RECORDS OF EIGHTY (80) EBCDIC CHARACTERS.

OPTIONS: THE FOLLOWING OPTIONS MAY BE SPECIFIED:

HERE TYPES THE DATA SET ON CONSOLE PRINTER, THE DEFAULT CONDITION.

ERASE IF YOU WISH THE DATA SET TO BE ERASED FROM THE DISK AFTER IT HAS BEEN TYPED.

LIMITS IF YOU WISH TO TYPE ONLY A PROTION OF EACH DATA SET, YOU MAY ENTER A COLLECTION OF THREE LIMITS, AS FOLLOWS:

THE FIRST NUMBER INDICATES THE NUMBER OF CHARACTERS, STARTING WITH THE CHARACTER IN COLUMN ONE OF EACH RECORD, THAT ARE TO BE TYPED FROM EACH RECORD. THE DEFAULT IS 80, THE ENTIRE RECORD.

THE SECOND NUMBER INDICATES THE INDEX OF THE FIRST RECORD OF THE DATA SET TO BE TYPED. THE DEFAULT IS ONE, THE FIRST RECORD OF THE DATA SET.

THE THIRD NUMBER INDICATES THE NUMBER OF THE INDEX OF THE LAST RECORD OF THE DATA SET TO BE TYPED. THE DEFAULT IS THE LAST RECORD OF THE DATA SET.

NOTES: SEE PRINT COMMAND FOR EXAMPLES.

UNCATALOGUE DSNAME

FUNCTION: TO MOVE A DATA SET FROM A LIBRARY TO SYSTEM WORKING STORAGE AREA, THEN TO REMOVE THE CORRESPONDING ENTRY FROM THE LOCATION EQUIVALENCE TABLE.

PARAMETER:

DSNAME — THE NAME OF THE DATA SET TO BE MOVED. THE NAME WILL NO LONGER BE IN THE USER'S LIBRARY CATALOGUE AFTER THE COMMAND HAS EXECUTED, AND THEREFORE THE NAME MAY BE GIVEN TO SOME OTHER DATA SET LATER.

NOTE: THE DATA SET IN WORKING STORAGE CANNOT BE MANIPULATED BY CYTOS UNTIL IT HAS BEEN EITHER CATALOGUED OR RECATALOGUED UNDER A NEW NAME -- THAT IS, UNTIL IT ONCE AGAIN HAS AN ENTRY IN A LIBRARY DIRECTORY.

UNPERMIT CARTRIDGE_ID ACCESS PERMITTEES

FUNCTION: REMOVE OR REVOKE ACCESS TO DATA SETS ON A SPECIFIED VOLUME FROM USERS DESCRIBED BY 'PERMITTEE'. UNPERMIT IS THE COMPLEMENTARY FUNCTION TO THE PERMIT COMMAND.

PARAMETERS:

CARTRIDGE — THE CARTRIDGE_ID OF THE VOLUME FOR WHICH PERMISSION STATUS IS TO BE ALTERED.

ACCESS — SEE PERMIT FOR TYPES OF ACCESS.

PERMITTEE — SEE PERMIT COMMAND.

EXAMPLE:

```
==* UNPERMIT 1011 WRITE AUTHOR
    AUTHOR   R   C E
    PROJECT  R W
    OTHER
***END OF PERMIT LIST***
```

VOID PROCESSOR

FUNCTION: TO REMOVE ONE OF THE IBM PROCESSORS FROM THE SYSTEM AREA OF THE CURRENTLY DEFINED SYSTEM CARTRIDGE.

PROCESSOR: ONE OF THE FOLLOWING:
ASSEMBLER, TO DELETE THE IBM MACRO ASSEMBLER;
FORTRAN, TO DELETE THE FORTRAN COMPILER;
RPG, TO DELETE THE RPG COMPILER; OR
COBOL, TO DELETE THE COBOL COMPILER.

NOTES: (1) A PROCESSOR CANNOT BE REMOVED FROM A 'SATELLITE' CARTRIDGE; TO DO SO, YOU MUST REDEFINE THE SYSTEM CARTRIDGE TO BE THAT SATELLITE CARTRIDGE FROM WHICH YOU WISH TO DELETE THE PROCESSOR. THIS MAY BE ACCOMPLISHED BY USING THE MOUNT COMMAND, SO LONG AS CYTOS ALSO RESIDES ON THE VOLUME FROM WHICH THE PROCESSOR IS TO BE REMOVED.

(2) IF CYTOS DOES NOT RESIDE ON THE CARTRIDGE FROM WHICH THE PROCESSOR IS TO BE REMOVED, YOU MUST USE THE *DEFINE VOID FEATURE OF THE DISK UTILITY PROGRAM IN BATCH MODE.

(3) ONCE A PROCESSOR HAS BEEN REMOVED, IT CANNOT BE RELOADED ONTO THE VOLUME EXCEPT BY MEANS OF THE "INITIAL LOAD" PROCEDURE DESCRIBED IN THE "PROGRAMMER AND OPERATOR'S GUIDE."

WORKON DSNAME

FUNCTION: PLACE A COPY OF A DATA SET IN SYSTEM WORKING STORAGE.

DSNAME: ANY DATA SET (TYPE .S, .O, OR .L) MAY BE COPIED INTO SYSTEM WORKING STORAGE. WORKING STORAGE INDICATORS IN DCOM ARE UPDATED TO REFLECT THE NEW CONTENTS OF WORKING STORAGE. ANYTHING PREVIOUSLY IN WORKING STORAGE WILL BE DESTROYED.

NOTE: IF THE DATA SET TO BE COPIED BELONGS TO ANOTHER USER, PERMISSION TO BOTH READ AND COPY MUST HAVE BEEN GRANTED BY THE OWNER.

```
XF  FILE_PSEUDONYM  DSNAME  <CART_ID>
```

FUNCTION: ESTABLISH CORRESPONDENCE BETWEEN A FILE-PSEUDONYM USED IN A PORGRAM AND THE ACTUAL DATA SET TO BE READ OR WRITTEN FOR USERS OF DNA'S XBASE PACKAGE. IF USING XBASE MULTIPLE ENTRIES CAN BE ENTERED FOR THE SAME FILE_PSEUDONYM TO ESTABLISH XBASE CONCATENATED FILES.

PARAMETERS:

FILE_PSDNYM THE DECIMAL FILE NUMBER USED TO IDENTIFY THE FILE IN THE PROGRAM (E.G. THE FORTRAN DEFINE FILE IDENTIFICATION).

DSNAME ANY VALID DATA SET NAME, INCLUDING "#" WHICH INDICATES A WORKING STORAGE FILE.

CART_ID IF GIVEN, INDICATES THE VOLUME THE DATA SET IS LOCATED ON.

EXAMPLE: ==> XF 1 DATAX 1111

WOULD SET UP A FILE CORRESPONDENCE BETWEEN DEFINE FILE 1 AND DATA SET DATAX ON CARTRIDGE 1111.

NOTE: (1) USING XF ALLOWS USERS TO SET UP CORRESPONDENCES ON UNMOUNTED CARTRIDGES.

(2) THERE IS NO DATA SECURITY CHECK MADE AS THERE IS WITH THE NAMEFILE COMMAND.

(3) SEE WRITE-UP IN DNA'S XBASE MANUAL, USING XBASE UNDER CYTOS.

```
ZAP  PHID  ADDR  VERIFY  (NEWVAL, ...)
```

FUNCTION: PATCH SYSTEM PHASES FROM THE TERMINAL.

ALIAS: Z

PARAMETERS:

PHID — PHASE IDENTIFICATION (HEXADECIMAL). TO SPECIFY A LOGICAL DRIVE "PHID" MUST CONSIST OF 4 HEXADECIMAL DIGITS THE FIRST DIGIT BEING THE LOGICAL DRIVE.

ADDR — PROGRAM OFFSET (THE LOCATION IN THE LISTING).

VERIFY — THE VALUE OF THE FIRST WORD TO BE CHANGED -- USED TO MAKE SURE YOU ARE AT THE CORRECT LOCATION.

NEWVALS — A LIST OF UP TO 20 HEXADECIMAL VALUES TO BE STORED IN THE LOCATIONS 'ADDR' THRU 'ADDR+19'

EXAMPLE: ==> ZAP 6E 50A 4C00 (4C80 69E)

WOULD CHANGE LOCATIONS 50A AND 50B TO THE VALUES IN THE REPLACEMENT COLLECTION ASSUMING THAT THE VERIFICATION VALUE MATCHES WHAT IS ALREADY THERE. PHASE 6E CORRESPONDS TO THE MONITOR CONTROL RECORD ANALYZER.

TO PUT THE SAME ZAP ON LOGICAL ONE YOU WOULD PUT IN THE FOLLOWING:

==> ZAP 106E 50A 4C00 (4C80 69E)

MESSAGE: ZP04 OLD=XXXX, GIVEN=XXXX

INDICATES AN INVALID VERIFICATION. THE MESSAGE DISPLAYS THE OLD (EXISTING) VALUE AND THE VALUE YOU ATTEMPTED TO SUBSTITUTE.

```
-----------------------------------------------------------
|                                                         |
|          TELL  AREANO  'MESSAGE'  <OPTIONS>             |
|                                                         |
-----------------------------------------------------------
```

FUNCTION: SEND MESSAGE TO THE CONSOLE TYPEWRITER OF ONE OR MORE TSO AREAS.

PARAMETERS:

AREANO — THE AREA NUMBER OF THE TERMINAL TO RECEIVE YOUR MESSAGE (AREA NOS. ARE -1 TO 15) OR A COLLECTION OF AREA NUMBERS. USING AN AREA NUMBER OF 99 WILL SEND YOUR MESSAGE TO ALL AREAS.

MESSAGE — THE MESSAGE TO BE SENT.

OPTIONS

ASK — WAIT FOR AN ANSWER FROM THE AREA.

INTERRUPT — CAUSES PRESENT OPERATION AT AN AREA BEING TOLD TO BE IMMEDIATELY INTERRUPTED. (SEE NOTES BELOW.)

OVERRIDE — OVERRIDE AREA IF HE IS IN NO-TELL MODE.

SECONDS(N) — N IS THE NUMBER OF SECONDS TO WAIT FOR A RESPONSE IF YOU USED THE ASK OPTION. IF NOT SPECIFIED, 300 SECONDS WILL DEFAULT.

EXAMPLE:

==> TELL -1 'MOUNT SCRATCH TAPE ON DRV 1' ASK

WOULD SEND THE MESSAGE "MOUNT SCRATCH TAPE ON DRV 1" TO AREA -1 AND WAIT FOR AN ANSWER.

==> TELL (1, 2) 'GOOD MORNING'

WOULD SEND THE MESSAGE "GOOD MORNING" TO AREAS 1 & 2.

NOTES: INTERRUPT NOT USED AS AN OPTION TO TELL

A. RECEIVING TERMINAL WILL RECEIVE THE TELL MESSAGE AFTER THE CURRENT LINE IF I/O IS IN PROGRESS, IF NO I/O IS IN PROGRESS HE WILL RECEIVE IT IMMEDIATELY.

INTERRUPT USED AS AN OPTION TO TELL

A. IF RECEIVING TERMINAL IS CURRENTLY PRINTING, HE WILL RECEIVE THE TELL MESSAGE AFTER THE CURRENT LINE IS TYPED.
B. IF RECEIVING TERMINAL IS INPUTTING DATA, HE WILL RECEIVE THE TELL MESSAGE IMMEDIATELY AND WILL HAVE TO RE-ENTER THE CURRENT LINE OF DATA HE WAS ENTERING.

```
-----------------------------------------------------------
|                                                         |
|        CIZAP  PROG  ADDR  VERIFY  (NEWVAL,...)          |
|                                                         |
-----------------------------------------------------------
```

FUNCTION: PATCH A CORE-IMAGE (.L) PROGRAM.

PARAMETERS:

PROG — NAME OF THE .L PROGRAM TO BE PATCHED.

ADDR — ADDRESS WITHIN THE .L PROGRAM.

VERIFY — THE VALUE OF THE FIRST WORD TO BE CHANGED -- USED TO MAKE SURE YOU ARE AT THE CORRECT LOCATION.

NEWVALS — A LIST OF UP TO 20 HEXIDECIMAL VALUES TO BE STORED IN THE LOCATIONS "ADDR" THRU "ADDR+19".

EXAMPLE: ==> CIZAP XPROG.L 50A 4C20 (4C00, 4C08)

WOULD PATCH THE VALUE IN CORE IMAGE PROGRAM XPROG AT ADDRESS 50A FROM 4C20 TO 4C00 AND THE NEXT LOCATION TO 4C08.

```
-----------------------------------------------------------
|                                                         |
|                SYSDUMP  PHID                            |
|                                                         |
-----------------------------------------------------------
```

FUNCTION: DUMP A SYSTEM PHASE TO THE PRINTER.

PARAMETERS:

PHID — THE HEXIDECIMAL ID NUMBER OF THE PHASE TO BE DUMPED. THE HIGH ORDER BITS (0-3) CONTAIN THE LOGICAL DRIVE NUMBER.

```
-----------------------------------------------------------
|                                                         |
|          STRIKE  (DSNAME,...)  <OPTIONS>                |
|                                                         |
-----------------------------------------------------------
```

FUNCTION: ERASE AND UNCATALOGUE ONE OR MORE DATA SETS REPLACING THEM WITH A "2DUMY" ENTRY WITHOUT PACKING UP THE LET TABLE.

PARAMETERS:

DSNAME — YOU MAY SPECIFY ANY TYPE OF DATA SET, INCLUDING '*' WHICH WILL CAUSE ALL TYPES FOR THE GIVEN NAME TO BE REMOVED.

OPTIONS

PACK — PACKS LET TABLES ON ALL MOUNTED CARTRIDGES.

EXAMPLES:

==> STRIKE SFIO.O

WOULD CAUSE SFIO.O TO BE REMOVED FROM THE LET TABLE, NO PACK WOULD OCCUR.

==> STRIKE SFIO.O PACK

WOULD REMOVE SFIO.O FROM THE LET TABLE AND THEN THE LET'S OF ALL MOUNTED CARTRIDGES WOULD BE PACKED. THIS EXAMPLE IS THE EQUIVALENT OF THE "ERASE" COMMAND.

==> STRIKE (PRNT1.O(7001), FILE1.S(7002)) PACK

WOULD REMOVE PRNT1.O FROM CARTRIDGE 7001, FILE1.S FROM CARTRIDGE 7002 AND IT WOULD THEN PACK THE LET'S OF ALL MOUNTED CARTRIDGES.

10. TIPS AND TECHNIQUES

THE INFORMATION IN THIS SECTION WILL HELP YOU TO USE THE CYTOS SYSTEM MORE EFFICIENTLY. YOU SHOULD HAVE A BASIC UNDERSTANDING OF BOTH THE CYTOS COMMAND LANGUAGE AND THE DISK MONITOR SYSTEM TO TAKE FULL ADVANTAGE OF THE VARIOUS SHORTCUTS PRESENTED HERE.

A. ON USING THE 'FIXED AREA' FACILITY

THE 'FIXED-AREA' UNDER CYTOS CAN BE USED IN THE SAME WAY AS UNDER THE DISK MONITOR SYSTEM IN BATCH MODE: FILES WHOSE LOCATION CANNOT CHANGE FOR ONE REASON OR ANOTHER MAY BE PERMANENTLY ANCHORED TO A FIXED POSITION ON A DISK VOLUME. HOWEVER THERE ARE SEVERAL POINTS WHICH YOU SHOULD KEEP IN MIND WHEN DECIDING UPON THE DESIRABILITY OF INCLUDING A FIXED AREA ON ANY OF YOUR CYTOS VOLUMES:

1. IF CYTOS IS TO BE LOADED ONTO YOUR MASTER CARTRIDGE VIA A SYSTEM RELOAD (AS OPPOSED TO AN INITIAL LOAD), A FIXED AREA CANNOT BE PRESENT AT THE TIME OF THE RELOAD.

2. IT IS NOT NECESSARY UNDER CYTOS TO PLACE FILES IN THE FIXED AREA IF THEY ARE REFERENCED BY CORE IMAGE PROGRAMS VIA THE DEFINE FILE TABLE; WHEN THE LOAD MODULE IS BROUGHT INTO CORE FOR EXECUTION, YOU WILL BE PROMPTED FOR THE NAME OF A DATA SET THAT IS TO BE USED FOR EACH OF THE SYMBOLIC FILES REFERENCED BY THE PROGRAM. THE CURRENT LOCATION AND SIZE OF THE FILE WILL THEN BE PLACED IN THE LOAD MODULE HEADER. IF REFERENCES HAD BEEN NAMEFILED PREVIOUSLY, THAT INFORMATION IS USED IN SETTING UP THE PROGRAM HEADER. YOU SHOULD REMEMBER THAT A LOAD MODULE OUGHT NOT BE REMOVED TO CARDS IF IT CURRENTLY CONTAINS REFERENCES IN ITS HEADER TO NON-SCRATCH FILES ON YOUR VOLUME AND IF IT IS SUBSEQUENTLY TO BE RUN ON A NON-CYTOS SYSTEM; THE REFERENCES TO DATA SETS ARE NOT SEARCHED AT EXECUTION TIME UNDER NORMAL DMS BATCH MODE UNLESS APPROPRIATE *FILES CARDS ARE INCLUDED FOR THE LOADING OF THE PROGRAM. THE PROGRAM COULD THEREFORE WRITE INTO EXTRANEOUS LOCATIONS ON THE NON-CYTOS VOLUME WHEN EXECUTED IN BATCH MODE.

3. USE THE FIXLIB COMMAND TO CREATE OR MODIFY THE SIZE OF A FIXED AREA ON ANY MOUNTED VOLUME.

4. THE SIZE OF ALL SOURCE AND LOAD MODULES MOVED TO THE FIXED AREA BY DMS DUP IS ASSUMED TO BE IN INTEGRAL SECTORS. BECAUSE CYTOS SOURCE FILES (AS CREATED BY THE EDITOR OR READ COMMAND) ARE ONLY AS LONG AS THE ACTUAL INFORMATION IN THE DATA SET, THE SAME ASSUMPTION DOES NOT HOLD UNDER CYTOS. IF YOU MOVE A CYTOS .S DATA SET (BY COPY, READ, OR WHATEVER MEANS) TO THE FIXED AREA AND HAVE NOT PREVIOUSLY SET ROUND TO PAD OUT THE LAST SECTOR (IF THE DATA SET IS NOT OF INTEGRAL SECTOR SIZE), THEN

THE DATA SET MOVED INTO THE FIXED AREA AND ALL SUBSEQUENT DATA SETS MOVED MAY CONTAIN ERRONEOUS INFORMATION, BECAUSE THE INFORMATION IS MOVED FROM WORKING STORAGE OR THE USER AREA A FULL SECTOR AT A TIME.

5. ANY CYTOS COMMAND THAT STORES INFORMATION INTO A DATA SET OR CREATES A NEW DATA SET IN THE FIXED AREA (AS OPPOSED TO REFERRING TO AN OLD DATA SET TO BE READ) MUST INCLUDE IN THE ACCOMPANYING DATA SET NAME PARAMETER THE FIXED LIBRARY SPECIFICATION -- THE NAME SUBSCRIPTED WITH A COLON AND THE CARTRIDGE_ID (FOR EXAMPLE, FXDATA.S(:2222)).

B. ON INTERRUPTING AND RESUMING PROGRAMS

UNDER CYTOS, YOU MAY INTERRUPT ALL USER PROGRAMS AND MOST SYSTEM COMMANDS BY PRESSING THE 'INT REQ' KEY ON THE SYSTEM CONSOLE KEYBOARD. THE FIRST 4K OF CORE IS IMMEDIATELY SAVED ON THE CORE IMAGE BUFFER AND CYTOS RETURNS TO COMMAND LEVEL TO PROMPT FOR ANOTHER COMMAND. CORE ABOVE 4K REMAINS UNTOUCHED UP TO THIS POINT. SOME CYTOS COMMANDS USE ONLY THE FIRST 4K OF CORE, AND MAY BE ISSUED WITHOUT DESTROYING THE CONTENTS OF UPPER CORE. OTHER COMMANDS REQUIRE CORE STORAGE BEYOND HEXADECIMAL LOCATION 1000 (E.G., READ AND LINKEDIT); THEREFORE, IF YOU INTEND TO RESUME PROCESSING OF THE INTERRUPTED PROGRAM, UPPER CORE MUST FIRST BE SAVED USING THE CHECKPOINT COMMAND. THOSE COMMANDS THAT RUN IN 4K ARE LISTED IN APPENDIX A.

WHEN YOU CHECKPOINT A PROGRAM, THE FIRST 4K OF THE INTERRUPTED PROGRAM IS COPIED FROM THE CORE IMAGE BUFFER TO THE CHECKPOINT DATA SET (WHICH MUST ALREADY EXIST, BECAUSE IT CANNOT BE CREATED WITHOUT DESTROYING THE CURRENT CONTENTS OF UPPER CORE). UPPER CORE IS THEN WRITTEN DIRECTLY AFTER THE FIRST 4K, STARTING IN SECTOR 17 OF THE CHECKPOINT DATA SET. AFTER YOU HAVE GIVEN THE CHECKPOINT COMMAND, YOU MAY ISSUE THE RESUME COMMAND RIGHT AWAY, BECAUSE UNLESS YOU HAVE PRESSED 'INT REQ' AGAIN, THE CONTENTS OF UPPER CORE AND THE CORE IMAGE BUFFER ARE THE SAME AS WHEN THE PROGRAM WAS INTERRUPTED. OR YOU MAY WISH TO ISSUE OTHER CYTOS COMMANDS THAT USE UPPER CORE, AND AT A LATER TIME GIVE THE RESTART COMMAND TO RESUME PROCESSING.

RESTART RESTORES THE PROGRAM TO ITS STATUS AT THE TIME THE CHECKPOINT COMMAND WAS GIVEN; UPPER CORE IS READ FROM THE CHECKPOINT/RESTART DATA SET AND THE FIRST 4K OF THE INTERRUPTED PROGRAM IS COPIED BACK TO THE CORE IMAGE BUFFER. SPECIFYING RESUME AS AN OPTION TO THE RESTART COMMAND, OR ISSUING A SEPARATE RESUME COMMAND, CAUSES CYTOS TO OVERLAY ITSELF WITH THE FIRST 4K OF THE INTERRUPTED PROGRAM FROM THE CORE IMAGE BUFFER. THEREFORE, RESTART IS THE COMPLEMENTARY FUNCTION OF CHECKPOINT, AND RESUME IS THE COMPLEMENTARY FUNCTION OF PRESSING THE 'INT REQ' KEY.

BETWEEN CHECKPOINT AND RESUME NEITHER THE SIZE AND LOCATION OF SYSTEM WORKING STORAGE NOR THAT PART OF ITS CONTENTS REFERENCED BY THE INTERRUPTED PROGRAM CAN DIFFER. IF YOU INTEND TO ISSUE COMMANDS THAT REQUIRE THE USE OF WORKING STORAGE BETWEEN THE TIME YOU CHECKPOINT AND RESTART, YOU MUST SAVE THE CONTENTS REFERENCED BY THE INTERRUPTED PROGRAM BY ISSUING A NEWSPACE OF SUFFICIENT SIZE. TO DO SO YOU MUST BE FAMILIAR WITH HOW THE INTERRUPTED PROGRAM WAS USING WORKING STORAGE WHEN YOU PRESSED THE 'INT REQ' KEY. JUST BEFORE YOU RESTART, YOU MUST ERASE THE NEWSPACED DATA SET SO AS TO RESTORE THE CONTENTS OF WORKING STORAGE TO THE ORIGINAL STATUS. IF YOU HAVE A MULTIPLE-DRIVE SYSTEM, YOU CAN TEMPORARILY ASSIGN WORKING STORAGE TO A DIFFERENT VOLUME, THEN REASSIGN THE ORIGINAL WORKING STORAGE BEFORE YOU RESTART.

NO PROGRAM, WHETHER USER OR SYSTEM, CAN BE RESUMED IF IT WAS ORIGINALLY INTERRUPTED WHILE EXECUTING AN INSTRUCTION EITHER IN THE RESIDENT MONITOR OR IN THE DISK I/O ROUTINE WHILE AT MAINLINE LEVEL; THIS CODE MUST BE USED TO ROLL OUT THE FIRST 4K TO THE CORE IMAGE BUFFER, SO ITS STATUS WILL BE THAT OF THE INSTANT THE WRITE TO DISK TOOK PLACE. THE ENTRY POINT TO THE DISK ROUTINE WILL HAVE BEEN DESTROYED BY THE CALL FROM $DUMP. THERE IS NO SPACE IN THE RESIDENT INTERRUPT LEVEL FOUR SUBROUTINE TO CHECK WHETHER THE INTERRUPT OCCURRED FROM A LOCATION OUTSIDE THESE SPECIAL AREAS. HOWEVER, IF YOU WANT TO MAKE SURE THAT ONLY VALID INTERRUPTS ARE HANDLED IN THE MANNER DESCRIBED AND INVALID INTERRUPTS ARE IGNORED, YOU CAN USE THE ILSX4 SUBROUTINE SUPPLIED WITH CYTOS, OR CODE YOUR OWN INTERRUPT LEVEL FOUR SUBROUTINE BASED UPON THE CYTOS VERSION.

ALL SYSTEM PROGRAMS USE THE MONITOR'S LEVEL FOUR INTERRUPT SUBROUTINE. ANY CORE LOAD BUILT AS A USER PROGRAM WILL ALSO USE THE MONITOR'S SUBROUTINE UNLESS ONE OF THE FOLLOWING CONDITIONS IS IN EFFECT:

1. YOU HAVE REPLACED THE DUMMY ILS04 SUBROUTINE WITH YOUR OWN VERSION WITH A SUB-TYPE OF ZERO. THIS WILL BE USED IF YOU DO NOT SPECIFY THAT THE SPECIAL ILSX4 SUBROUTINE BE USED WHEN YOUR LOAD MODULE IS BUILT.

2. YOU SPECIFY THE USE OF AN ILSX4. IF IBM'S ILSX4 IS USED INSTEAD OF THE CYTOS ILSX4 PROVIDED AT SYSGEN TIME, NO USER WRITTEN PROGRAMS WILL BE RESUMABLE.

YOU SHOULD THEREFORE SPECIFY EITHER THE ILSX4 PROVIDED WITH CYTOS, OR SPECIFY NOTHING, IF YOU WISH YOUR INTERRUPTABLE PROGRAMS TO BE RESUMABLE AS WELL.

ANOTHER METHOD OF INSURING THAT PROGRAMS ARE NOT INTERRUPTED WITHOUT LATER BEING RESUMABLE IS TO CODE A PAUSE OR WAIT INTO THE PROGRAM AT THE POINT YOU WISH TO SAVE THE INTERRUPTED STATUS. ALTERNATIVELY, YOU MAY CAUSE A RECORD TO BE WRITTEN TO A NON-READY I/O DEVICE; THE PRE- AND POST-OPERATIVE ERROR TRAPS, AS WELL AS THE PROGRAM STOP KEY SUBROUTINE, ARE EXEMPT FROM THE AFOREMENTIONED RESTRICTION ON RESUMING PROGRAMS INTERRUPTED FROM WITHIN THE RESIDENT MONITOR. YOU CAN BE SURE OF THE RESUMABILITY OF THE PROGRAM BY MANUALLY EXAMINING THE RETURN ADDRESS FROM ANY ONE OF THESE LOW-CORE SUBROUTINES. ALTHOUGH HIGHLY INFLEXIBLE, THIS METHOD WILL INSURE RESUMABILITY.

ONE OTHER FACTOR TO KEEP IN MIND: YOU SHOULD NOT ATTEMPT TO CHECKPOINT/RESTART A PROGRAM YOU KNOW TO BE USING THE WRITE IMMEDIATE FUNCTION OF EITHER THE DISK1 OR DISKN I/O ROUTINES. THE CURRENT POSITION OF THE DISK ARM IS NOT EXAMINED BEFORE THE WRITE, SO RESUMING IN SUCH AN INSTANCE MIGHT CAUSE A DISK-WRITE TO AN UNINTENDED LOCATION ON THE VOLUME WHICH COULD BE POTENTIALLY DISASTROUS.

C. ON DYNAMIC VOLUME MOUNTING

MOUNTING OF VOLUMES IS CONTROLLED UNDER CYTOS BY THE LOGON AND MOUNT COMMANDS. YOU MAY BE RUNNING PROGRAMS THAT CALL THE IBM SYSUP SUBROUTINE TO MOUNT DIFFERENT VOLUMES DURING THE EXECUTION OF YOUR PROGRAM (AS WHEN YOU MUST PROCESS A MULTI-VOLUME FILE ON A ONE DRIVE CONFIGURATION). SINCE CYTOS CANNOT CONTROL SUCH VOLUME SWITCHING IN THE MIDDLE OF USER PROGRAMS, IT MUST ASSUME THAT IF AT THE TERMINATION OF A USER COMMAND THE VOLUMES MOUNTED ARE DIFFERENT FROM THOSE MOUNTED AT THE TIME THE USER PROGRAM WAS SET INTO EXECUTION, THEN THE MASTER CARTRIDGE FROM WHICH CYTOS IS RUNNING AT THE END MAY IN FACT BELONG TO ANOTHER PERSON THAN THE PERSON LOGGED ON PREVIOUSLY. IF THE VOLUMES DO DIFFER, YOU WILL BE PROMPTED TO LOGON TO THE SYSTEM AGAIN AND IDENTIFY YOURSELF WITH USERID, PROJID, AND PASSWORD.

REMEMBER THAT IF THE MASTER CARTRIDGE MOUNTED DYNAMICALLY VIA SYSUP IS NOT A CYTOS SYSTEM CARTRIDGE, THERE WILL BE NO WAY TO RETURN TO CYTOS COMMAND LEVEL AFTER THE USER PROGRAM HAS TERMINATED EXECUTION. YOU WILL HAVE TO REMOUNT A CYTOS MASTER CARTRIDGE VIA THE '// JOB' CONTROL RECORD IN ORDER TO RUN UNDER CYTOS AGAIN.

D. ON OPTIMIZING DISK VOLUME SPACE

IF YOU ARE SHORT ON DISK SPACE, THE FOLLOWING TECHNIQUES MAY BE OF SOME ASSISTANCE:

1. MAINTAIN A CYTOS SYSTEM VOLUME THAT HAS ALL THE COMPILERS YOUR INSTALLATION USES, ALONG WITH THE CORE LOAD BUILDER AND YOUR PROGRAM AND SUBROUTINE LIBRARY. DEVELOP YOUR LOAD MODULES ON THIS VOLUME AND THEN MOVE THEM (IF YOU HAVE A MULTI-DRIVE CONFIGURATION) OR REMOVE AND RESTORE THEM (IF YOU HAVE ONLY THE DRIVE HOUSED IN THE CPU) TO OTHER CYTOS VOLUMES ON WHICH THE DATA SETS TO BE HANDLED BY THE PROGRAMS HAVE PREVIOUSLY BEEN PLACED. THE COMPILERS NEED NOT BE DUPLICATED ON EVERY CARTRIDGE.

2. WITH REMOVE AND RESTORE, BACKUP TO CARDS UNDER CYTOS IS EXTREMELY CONVENIENT AND EASY. USE THIS FACILITY LIBERALLY. WHEN YOU HAVE REMOVED A DATA SET, WRITE THE DECK CARD-COUNT ON TOP OF THE DECK SO THAT IT WILL BE EASY TO READ WHEN YOU ARE ISSUING THE RESTORE COMMAND.

3. IF YOU HAVE A MULTI-DRIVE SYSTEM, PLACE THE EXPLAIN FILE ON A SATELLITE CARTRIDGE THAT CAN BE MOUNTED ONLY AT THOSE TIMES WHEN YOU ARE LIKELY TO ENCOUNTER ERROR MESSAGES FOR THE FIRST TIME. THE EXPLAIN FILE IS LOCATED AUTOMATICALLY BY CYTOS; THE VOLUME ON WHICH IT RESIDES NEED BE MOUNTED ONLY AT THE TIME YOU ISSUE THE EXPLAIN COMMAND.

E. ON OPTIMIZING DISK ACCESS TIME

DISK ARM MOVEMENT CAN BE SIGNIFICANTLY REDUCED, ESPECIALLY ON A MULTI-DRIVE SYSTEM, IF THE FOLLOWING FACTORS ARE TAKEN INTO CONSIDERATION:

1. SEPARATE ONTO DIFFERENT VOLUMES AS MUCH AS POSSIBLE THE VARIOUS DISK AREAS:

   (A) THE CYTOS SYSTEM AND COMPILERS,
   (B) YOUR SOURCE PROGRAMS,
   (C) SYSTEM WORKING STORAGE.

CYTOS EDITOR UPDATING AND SOURCE PROGRAM COMPILATION WILL BE MUCH FASTER IF THESE THREE HAVE AS LITTLE VOLUME AFFINITY AS POSSIBLE.

2. WHEN COPYING A LARGE DATA SET, FIRST ASSIGN SYSTEM WORKING STORAGE TO THE VOLUME ON WHICH THE NEW DATA SET IS TO BE CATALOGUED.

3. ON A SINGLE DRIVE SYSTEM, EXECUTION TIME DISK ACCESS CAN BE REDUCED SIGNIFICANTLY IF YOUR LOAD MODULE AND THE DATA SETS THEY REFERENCE ARE ON A VOLUME WHICH DOES NOT INCLUDE THE IBM SUBROUTINE LIBRARY OR ANY OF THE VARIOUS COMPILERS.

4. PLACE DATA SETS THAT ARE ACCESSED AT EXECUTION TIME AS NEAR TO WORKING STORAGE (TOWARDS THE END OF THE USER'S LIBRARY) AS POSSIBLE, ESPECIALLY IF YOUR LOAD MODULE USES THE SUBROUTINE OVERLAY FEATURE. REMEMBER, CYTOS ALLOWS YOU ACCESS TO DATA SETS IN THE USER'S LIBRARY AT EXECUTION TIME, WHILE THE DISK MONITOR SYSTEM RUN IN BATCH MODE DOES NOT.

F. ON REPRODUCING BINARY DECKS

THE REPRODUCE COMMAND SHOULD BE USED ONLY TO REPRODUCE A SOURCE CARD FILE (ONE EBCDIC CHARACTER PER CARD COLUMN). IF YOU WANT TO REPRODUCE A BINARY DECK (54 WORDS PER CARD, 1 1/3 COLUMNS PER WORD), YOU CAN RESTORE THE DECK AS A .S DATA SET AND THEN IMMEDIATELY REMOVE IT TO CARDS AGAIN. THE EXACT CARD COUNT MUST BE GIVEN WITH THE RESTORE COMMAND. THE OUTPUT CARD COUNT WILL BE THE NEXT HIGHER INTEGRAL OF SIX CARDS -- THROW AWAY THE DIFFERENCE.

ANY BINARY CARD FILE MAY BE REPRODUCED IN THIS MANNER, NO MATTER WHAT TYPE OF DATA SET (.S, .O, OR .L) IT ORIGINALLY REPRESENTED ON THE DISK. THE ID STARTING IN COLUMN 73 OF THE NEW DECK WILL BE THE NAME YOU USED IN THE RESTORE AND REMOVE COMMANDS. THE LAST EIGHT COLUMNS OF THE CARD ARE NOT REPRODUCED, BUT RATHER RE-SEQUENCED.

G. ON MAINTAINING MACRO LIBRARIES

THE IBM MACRO UPDATE PROGRAM MAY BE RUN FROM THE TERMINAL UNDER CYTOS. TO ESTABLISH A MACRO LIBRARY, USE THE NEWSPACE COMMAND TO CREATE A SOURCE DATA SET. THEN ISSUE THE MACLIB HERE COMMAND TO INVOKE THE MACRO UPDATE PROGRAM. TYPE IN THE BUILD 'LIBNAME' COMMAND, FOLLOWED BY AN ENDUP COMMAND, AND YOUR MACRO LIBRARY WILL BE READY TO ACCEPT MACROS.

FOR EASE OF MAINTAINING A MACRO LIBRARY, YOUR ORIGINAL MACROS CAN BE ENTERED EITHER FROM CARDS VIA AN ASSEMBLY, USING THE SMAC OP-CODE OF THE MACRO ASSEMBLER, OR BY ASSEMBLING THESE SAME MACROS FROM A SOURCE DATA SET ON THE DISK. INPUT TO THE ASSEMBLER NEED CONTAIN ONLY THE MACRO DEFINITIONS AND AN ASSEMBLER END STATEMENT. THE DATA SET CONTAINING THE MACRO SOURCE CODE (AS OPPOSED TO THE MACRO LIBRARY ITSELF) CAN THEN BE MODIFIED USING THE CYTOS EDITOR. ALTERNATIVELY, YOU CAN USE THE COMMANDS PROVIDED BY THE MACRO UPDATE PROGRAM TO ADD, DELETE, AND CHANGE RECORDS DIRECTLY IN THE MACRO LIBRARY DATA SET.

MACRO LIBRARIES MAY BE REMOVED AND RESTORED TO THE DISK AS THEY ARE NEEDED. ACCORDING TO IBM SPECIFICATIONS, MACRO LIBRARIES MAY BE PLACED EITHER IN THE USER AREA OR IN THE FIXED AREA.

WHEN ENTERING MACRO UPDATE PROGRAM COMMANDS, THE COMMAND VERB (E.G., DELETE) MSUT BE LEFT-JUSTIFIED IN THE INPUT RECORD (UNLIKE CYTOS COMMANDS WHICH MAY BE ENTERED TOTALLY IN FREE FORMAT). FURTHERMORE, ANY ASSEMBLER INSTRUCTIONS ADDED VIA THE INSERT OR ADD FACILITY MUST BE ALIGNED AS THOUGH YOU WERE ENTERING IT ON A SOURCE CARD: TWENTY BLANKS MUST PRECEDE THE ASSEMBLER LABEL FIELD ON EACH RECORD ENTERED FROM THE TERMINAL IN MACRO UPDATE MODE.

IF YOU HAVE CONCATENATED MACRO LIBRARIES, YOU MUST SPECIFY THE NAME OF THE FIRST LIBRARY OF THE CHAIN AS THE MACLIB_NAME PARAMETER TO THE ASSEMBLE COMMAND FOR ANY OF THE MACROS IN THE CONCATENATED LIBRARIES TO BE AVAILABLE TO THE MACRO ASSEMBLER.

H. ON RUNNING IBM UTILITIES UNDER CYTOS

IBM MAINLINE UTILITY PROGRAMS MAY BE INVOKED UNDER CYTOS BY USING THE LOADGO COMMAND. SOME UTILITIES, SUCH AS DISC AND ID, REQUIRE INPUT PARAMETERS IN ORDER TO OPERATE. THESE MUST BE PROVIDED ON THE APPROPRIATE '*' CONTROL CARDS VIA THE SYSTEM INPUT DEVICE (CARD OR PAPERTAPE READER). THEY CANNOT BE ENTERED FROM THE TERMINAL BECAUSE THE UTILITIES SPECIFICALLY LOAD THE SYSTEM DEVICE SUBROUTINE FOR THE PRINCIPAL INPUT DEVICE. THE SYSTEM INPUT DEVICE IS NOT CONSIDERED TO BE THE TERMINAL UNLESS YOU HAVE PREVIOUSLY ENTERED THE '// TYP' MONITOR CONTROL RECORD, AFTER WHICH ALL MONITOR CONTROL RECORDS MUST BE ENTERED FROM THE TERMINAL UNTIL THE '// TEND' RECORD IS GIVEN.

I. ON EDITING LARGE FILES

THE EDITING OF LARGE FILES, ESPECIALLY ON CONFIGURATIONS WITH ONLY ONE DISK DRIVE, MAY POSE CERTAIN DIFFICULTIES IF THE SIZE OF SYSTEM WORKING STORAGE IS SMALLER THAN THAT OF THE FILE BEING EDITED. BY ISSUING AN UPDATE OR FILE COMMAND, YOU ASK THAT THE EDITOR MERGE ANY UNCHANGED RECORDS FROM THE EDITED DATA SET WITH NEW OR CHANGED RECORDS CREATED IN THE PROCESS OF EDITING. IF THE NEW MERGED TEXT IS LARGER THAN WORKING STORAGE, IT CANNOT BE COMPLETELY GENERATED; YOU WILL EITHER HAVE TO DELETE ENOUGH RECORDS TO BE ABLE TO CREATE A COMPLETE TEXT THAT WILL FIT IN THE UTILITY FILE, OR TERMINATE EDITING IN ORDER TO REMOVE OR ERASE DATA SETS FROM THE DISK SO AS TO MAKE WORKING STORAGE LARGER. ON A MULTI-DRIVE SYSTEM, THE PROBLEM CAN USUALLY BE SOLVED BY MOUNTING A SATELLITE CARTRIDGE TO WHICH SYSTEM WORKING STORAGE CAN BE ASSIGNED. YOU CAN THEN RE-EDIT THE ORIGINAL DATA SET FROM THE BEGINNING.

UNDER CERTAIN CIRCUMSTANCES, IT IS NOT EVEN NECESSARY TO CREATE A MERGED TEXT BEFORE UPDATING OR FILING. IF YOU OBSERVE THE FOLLOWING RULES, THE EDITED TEXT WILL BE 'FAST-FILED' -- ONLY THOSE RECORDS ACTUALLY CHANGED WILL BE WRITTEN BACK INTO THE EDITED DATA SET:

1. YOU MAY USE THE AFTER, BEFORE, CAPITALIZE, CHANGE, FOLD, OVERLAY, AND REPLACE COMMANDS TO CHANGE SPECIFIC RECORDS; YOU MAY NOT USE INSERT (OR INSERT-MODE), DELETE, OR INFL BECAUSE THESE CHANGE THE RELATIVE POSITIONS OF ALL SUBSEQUENT RECORDS IN THE DATA SET TEXT.

2. YOU CANNOT UPDATE AND MUST FILE BACK INTO THE ORIGINAL DATA SET (NOT INTO A NEW DATA SET); YOU MUST ISSUE THE FILE COMMAND BEFORE THE FIRST 'AUTOMATIC UPDATE' OCCURS.

3. YOU CANNOT USE THE LABEL COMMAND TO INSERT NEW ID-SEQUENCE NUMBERS IN THE DATA SET.

4. YOU CANNOT LEAVE THE EDITOR USING THE ATTN COMMAND AND THEN RESUME EDIT BEFORE GIVING THE FILE COMMAND.

IF ALL THESE CONDITIONS ARE MET, WORKING-STORAGE WILL BE USED ONLY TO CONTAIN THE CHANGED OR REPLACED RECORDS, AND THEREFORE NEED BE RELATIVELY SMALL COMPARED TO THE SIZE OF THE DATA SET YOU ARE EDITING.

IF THE UTILITY_FILE/WORKING_STORAGE IS SMALLER THAN THE DATA SET YOU INTEND TO EDIT, YOU WILL BE WARNED OF THIS CONDITION AT THE BEGINNING OF THE EDITING SESSION, BEFORE THE FIRST EDITOR PROMPT IS TYPED.

J. ON EDITING LARGE CARD FILES

SOMETIMES IT IS CONVENIENT TO EDIT LARGE CARD FILES ACCORDING TO THE TECHNIQUE PRESENTED HERE. YOU MUST READ THE FILE ONTO THE DISK BEFORE COMMENCING THE EDIT, BUT NEED NOT PUNCH OUT THE ENTIRE UPDATED FILE TO CARDS AGAIN. THIS PROCEDURE SAVES CONSIDERABLE TIME (AND CARDS) IF YOU INTEND TO CHANGE ONLY A VERY SMALL PERCENTAGE OF THE RECORDS IN THE CARD FILE.

EDIT THE FILE NOW ON DISK AS YOU WOULD ANY OTHER SOURCE DATA SET, ISSUING ANY OF THE COMMANDS (EXCEPT INFL) NECESSARY TO CHANGE THE VARIOUS RECORDS. KEEP A ROUGH COUNT OF THE NUMBER OF RECORDS YOU HAVE CHANGED. WHEN YOU HAVE FINISHED EDITING, ISSUE THE QUIT COMMAND INSTEAD OF THE FILE OR UPDATE COMMAND. WHEN AT CYTOS COMMAND LEVEL, GIVE A NEWSPACE COMMAND FOR A DATA SET LARGE ENOUGH TO INCLUDE THE TOTAL NUMBER OF RECORDS YOU HAVE CHANGED. THEN PUNCH THIS DATA SET TO CARDS, INTERPRET THE PUNCHED CARDS, AND INSERT THEM INTO THE ORIGINAL CARD FILE MANUALLY. RECORDS DELETED DURING EDITING MUST ALSO BE REMOVED MANUALLY, OF COURSE. TO EMPLOY THIS TECHNIQUE SUCCESSFULLY, IT IS HIGHLY RECOMMENDED THAT YOUR SOURCE CARD FILE HAVE SOME SEQUENCE IDENTIFICATION, UNLESS YOU ARE THOROUGHLY FAMILIAR WITH ITS CONTENTS.

K. ON DEFINING COMPLEX EDITOR PROCEDURES

SOME EDITOR COMMANDS ARE REPETITIVE BY NATURE, E.G., CHANGE. BY PLACING THEM IN A SIMPLE DO COMMAND COLLECTION, YOU CAN CHANGE EVERY OCCURRENCE OF ONE CHARACTER STRING IN A DATA SET TO SOME OTHER STRING. DEFINE-ING PROCEDURES ALLOWS YOU TO ESTABLISH MORE COMPLEX SETS OF EDITOR OPERATIONS, PROVIDING FOR MORE SELECTIVE CONTROL OF THE INFORMATION BEING CHANGED. BECAUSE EDITOR PROCEDURES CAN BE NESTED, YOU HAVE THE FURTHER CAPABILITY TO TEST EACH ONE SEPARATELY BEFORE COMBINING THEM INTO MORE COMPLEX FORMS.

SEVERAL USEFUL TECHNIQUES TO KEEP IN MIND WHEN DEFINING EDITOR PROCEDURES ARE LISTED HERE:

1. MAKE SURE THAT PROCEDURES ARE NOT DEFINED RECURSIVELY -- AN INNER PROCEDURE SHOULD CANN NEITHER ITSELF NOR ANY OUTER PROCEDURE.

APPENDIX A. CYTOS COMMAND ATTRIBUTES

THE FOLLOWING TABLE PRESENTS SYSTEM ATTRIBUTES OF ALL CYTOS SYSTEM COMMANDS. AN X INDICATES THAT THE ATTRIBUTE APPLIES TO THE PARTICULAR COMMAND. THE ABBREVIATED COLUMN HEADINGS STAND FOR:

- 4K - COMMAND RUNS IN FIRST 4K OF CORE STORAGE
- INT - COMMAND IS INTERRUPTIBLE (BY DEPRESSING 'INT REQ' KEY)
- RES - INTERRUPTED COMMAND IS RESUMABLE (VIA RESUME)
- WS - COMMAND USES SYSTEM WORKING STORAGE
- FPA - THE FILE PROTECT ADDRESS MAY BE CHANGED
- FPG - COMMAND MAY BE ISSUED PRIOR TO LOGON
- XMD - COMMAND ISSUABLE IN EXECUTE OR CARD-INPUT MODE
- OPR - COMMAND AVAILABLE ONLY TO SYSTEM OPERATOR. IF CYTOS/II

COMMAND	4K	INT	RES	WS	FPA	PLG	XMD	OPR
ALLOCATE		X	X		X		X	
ASSEMBLE		X	X	X	X		X	
ASSIGN	X	X					X	
BATCH	X					X		X
CATALOGUE		X		X	X		X	
CATPRINT		X	X				X	
CHECKPOINT	X						X	
CLEAR	X	X	X				X	
COBOL		X	X	X	X		X	
COMMA	X	X	X				X	
CONTINUE	X						X	
COPY		X	X	X	X		X	
DCOM	X	X	X				X	

COMMAND	4K	INT	RES	WS	FPA	PLG	XMD	OPR
DESCRIBE	X	X	X				X	
DUMP		X	X				X	
EDIT		X	X	X	X		X	
EJECT	X	X	X				X	
ERASE				X	X		X	
EXECUTE	X						X	
EXPLAIN	X	X	X			X	X	
FIXLIB					X		X	
FLUSH		X	X				X	
FORTRAN		X	X	X	X		X	
GRAPHICS	X	X	X				X	
INCLUDE	X	X	X				X	
JOIN	X	X	X			X	X	
KEYBOARD	X	X					X	
LINKEDIT		X		X	X		X	
LIST		X	X	X			X	X
LOADGO	X	X		X			X	
LOGOFF	X					X	X	
LOGON	X	X				X	X	
MACLIB		X	X				X	
MOUNT	X						X	
MOVE		X	X	X	X		X	
NAMEFILE	X	X	X				X	
NEWPASS	X	X	X				X	

COMMAND	4K	INT	RES	WS	FPA	PLG	XMD	OPR
NEWSPACE		X	X	X	X		X	
OVERLAY	X	X	X				X	
PERMIT	X	X	X				X	
PRINT		X	X		X		X	
PUNCH		X	X		X		X	
READ		X	X	X	X		X	
RECATALOGUE		X	X	X			X	
REFORMAT		X	X	X			X	X
REMOVE		X	X		X		X	
RENAME		X	X				X	
REPRODUCE		X	X	X			X	X
RESTART	X						X	
RESTORE		X	X	X	X		X	
RESUME	X						X	
RPG		X	X	X	X		X	
RUN	X	X	X	X			X	
SET	X	X	X				X	
SUBSTITUTE	X	X	X				X	
UNCATALOGUE		X		X	X		X	
UNPERMIT	X	X	X				X	
VOID					X		X	X
WHATFILE	X	X	X	X	X		X	
WORKON		X	X	X			X	
ZAP	X	X	X					X

APPENDIX B. DATA SET ATTRIBUTES, BY COMMAND

THE TABLE ON THE FOLLOWING PAGES PROVIDES INFORMATION ABOUT THE ATTRIBUTES OF ALL DATA SET NAME PARAMETERS FOR CYTOS SYSTEM COMMANDS. THE ABBREVIATED COLUMN HEADINGS ARE EXPLAINED IN THE KEY THAT FOLLOWS. EXPLANATIONS FOR THE NUMBERED NOTES ARE GIVEN AT THE END OF THE KEY.

KEY TO COLUMN HEADINGS:

DF - DEFAULT DATA SET TYPE
.D - SOURCE DATA SET ALLOWED
.O - OBJECT DATA SET ALLOWED
.L - LOAD DATA SET ALLOWED
.* - ALL (.S, .O, .L) ALLOWED
.T - INPUT FROM TERMINAL ALLOWED
.C - INPUT FROM PRINCIPAL I/O DEVICE ALLOWED, OR OUTPUT TO PRINCIPAL I/O DEVICE ALLOWED
.D - DUMMY OUTPUT ALLOWED
\# - SCRATCH DATA SET ALLOWED
() - COLLECTION OF NAMES ALLOWED
FX - DATA SET MAY BE IN FIXED AREA
RD - PERMISSION TO READ REQUIRED
WR - PERMISSION TO WRITE REQUIRED
CP - PERMISSION TO COPY REQUIRED
ER - PERMISSION TO ERASE REQUIRED (TO USE ERASE OPTION)
DS - DATA SET DISPOSITION:
  O = MUST BE OLD (EXISTING)
  N = MUST BE NEW (NOT YET EXISTING)
  E = EITHER OLD OR NEW
AO - AUTHOR'S DATA SET ONLY (I.E., NAME CANNOT BE MODIFIED BY A USERID)

NOTES INDICATED BY NUMBERS:

(1) SIZE OF CHECKPOINT/RESTART DATA SET MUST BE AS LARGE AS THE SIZE OF CORE, PLUS 3 SECTORS:

  04K = 16 SECTORS
  08K = 29 SECTORS
  16K = 55 SECTORS
  32K = 106 SECTORS

(2) SAVE/USE DATA SET MUST BE AT LEAST 26 "CARDS" LONG.

COMMAND	DATA SET	DF	.S	.O	.L	.*	.T	.C	.D	#	()	FX	RD	WR	CP	ER	DS	AO
ALLOCATE	DSNAME	.S	X								X	X		X			N	
ASSEMBLE	INDSNAME	.S	X				X	X				X	X				O	
	OUTDSNAME	.O		X				X	X					X			E	
	MACLIB	.S	X									X	X	X			O	
CATALOGUE	DSNAME	.S	X	X	X							X		X			E	
CHECKPOINT	DSNAME(1)	.S	X									X		X			O	
COBOL	INDSNAME	.S	X					X				X	X				O	
	OUTDSNAME	.O		X				X	X					X			E	
COPY	OLDDSNAME	.S	X	X	X	X					X	X	X		X		O	
	NEWDSNAME	.S	X	X	X	X					X	X		X			N	
DCIP	CARTN	.S	X								X	X		X			O	X
DESCRIBE	DSNAME	.S	X	X	X	X					X	X					E	
DUMP	DSNAME	.S	X	X	X	X					X	X					O	
EDIT	DSNAME	.S	X									X	X	X	X		E	
FILE	DSNAME	.S	X									X		X			E	
INFL	DSNAME	.S	X									X	X		X		O	
SAVE	DSNAME(2)	.S	X									X		X			O	
UPDATE	DSNAME	.S	X									X		X			E	
USE	DSNAME	.S	X									X	X				O	
UTILITY	DSNAME	.S	X									X		X			O	
ERASE	DSNAME	.S	X	X	X	X					X	X				X	O	
EXECUTE	DSNAME	.S	X									X	X				O	
FORTRAN	INDSNAME	.S	X				X	X				X	X				O	
	OUTDSNAME	.O		X				X	X					X			E	
GRAPHICS	PROG_NAME	.O		X									X				O	X
INCLUDE	PROG_NAME	.O		X									X				O	X
	SUBR_NAME	.O		X							X		X				O	X
LINKEDIT	INDSNAME	.O		X									X				O	
	OUTDSNAME	.L			X							X		X			N	
LOADGO	DSNAME	.O		X									X				O	
MOVE	OLDDSNAME	.S	X	X	X	X					X	X	X		X	X	O	
	NEWDSNAME	.S	X	X	X	X					X	X		X	X		E	
NAMEFILE	DSNAME	.S	X		X					X		X	X	X	X		E	
NEWSPACE	DSNAME	.S	X								X	X		X			N	

OVERLAY	PROG_NAME	.O		X									X				O	X
	SUBR_NAME	.O		X							X		X				O	X
PRINT	DSNAME	.S	X								X	X	X			X	O	
PUNCH	DSNAME	.S	X								X	X	X			X	O	
READ	DSNAME	.S	X								X	X		X			E	
RECATALOG	DSNAME	.S	X	X	X							X	X	X			O	
REMOVE	DSNAME	.S	X	X	X	X					X	X	X			X	O	
RENAME	OLDDSNAME	.S	X	X	X							X					O	X
	NEWDSNAME	.S	X	X	X							X					N	X
RESTART	DSNAME(1)	.S	X									X	X				O	
RESTORE	DSNAME	.S	X	X	X	X					X	X		X			N	
RPG	INDSNAME	.S	X				X	X				X	X				O	
	OUTDSNAME	.O		X				X	X					X			E	
RUN	DSNAME	.L			X							X	X				O	
SUBSTITUTE	REFRN_DSN	.O		X							X		X				O	X
	SUBST_DSN	.O		X							X		X				O	X
UNCATALOG	DSNAME	.S	X	X	X							X	X			X	O	
WHATFILE	DSNAME	.S	X							X		X	X	X			E	
WORKON	DSNAME	.S	X	X	X							X	X		X		O	

APPENDIX C. CYTOS/I WAIT CODES

CYTOS WAITS MAY BE RECOGNIZED BY THE PRESENCE OF A NON-ZERO RIGHT BYTE IN BOTH THE STORAGE BUFFER AND ARITHMETIC FACTOR INDICATOR DISPLAYS ON THE CONSULE DISPLAY PANEL. THE FOLLOWING TABLE INDICATES THE REASON FOR EACH WAIT AND THE APPROPRIATE RESPONSE.

MODULE	CODE	EXPLANATION	YOUR ACTION
@LG	/14	LOGOFF SHUTDOWN	POWER OFF
@DI	/20	ATTEMPTED OPEN, ALREADY OPEN	APAR, WITH DUMP
@DA	/21	ATTEMPTED OPEN, ALREADY OPEN	APAR, WITH DUMP
@DA	/22	ATTEMPTED OPEN, ALREADY CLOSED	APAR, WITH DUMP
@KB	/55	AWAITING KEYBOARD INPUT	RESPOND TO PROMPT
@WF	/60	INVALID SLET	APAR, WITH DUMP

APPENDIX D. DMS RESOURCES RESERVED FOR CYTOS

THE FOLLOWING LOCATIONS ARE USED BY CYTOS. THEY MAY CHANGE AT ANY GIVEN RELEASE UPDATE, AND ARE PRESENTED HERE ONLY TO HELP YOU AVOID CONFLICTS WITH ANY SYSTEM RESOURCES USED BY YOUR OWN SYSTEM PROGRAMS.

LOCATIONS IN COMMA (COMMUNICATIONS AREA):

REQUEST CURRENT LISTING OF RESIDENT IMAGE FOR MODIFICATION LEVEL AND VERSION OF CYTOS IN USE.

LOCATIONS IN SECTOR ZERO (ID SECTOR):

```
#USER /10 USERID (DOUBLE-WORD)
#PROJ /12 PROJID (DOUBLE-WORD)
#PROT /14 DATA SET PROTECTION MATRIX
#STAT /15 STATUS BITS
#PSWD /16 PASSWORD (DOUBLE-WORD)
```

SYSTEM PHASE IDS:			DECIMAL	HEX
CYTOS/I	V2-M11	DM2	256-284	100-11C
CYTOS/II	V2-M11	DM2	256-288	100-120
CYTOS/I	V2-M12	DM2	512-540	200-21C
CYTOS/II	V2-M12	DM2	512-544	200-220
APL OPTION			545-549	221-225