


Burroughs 

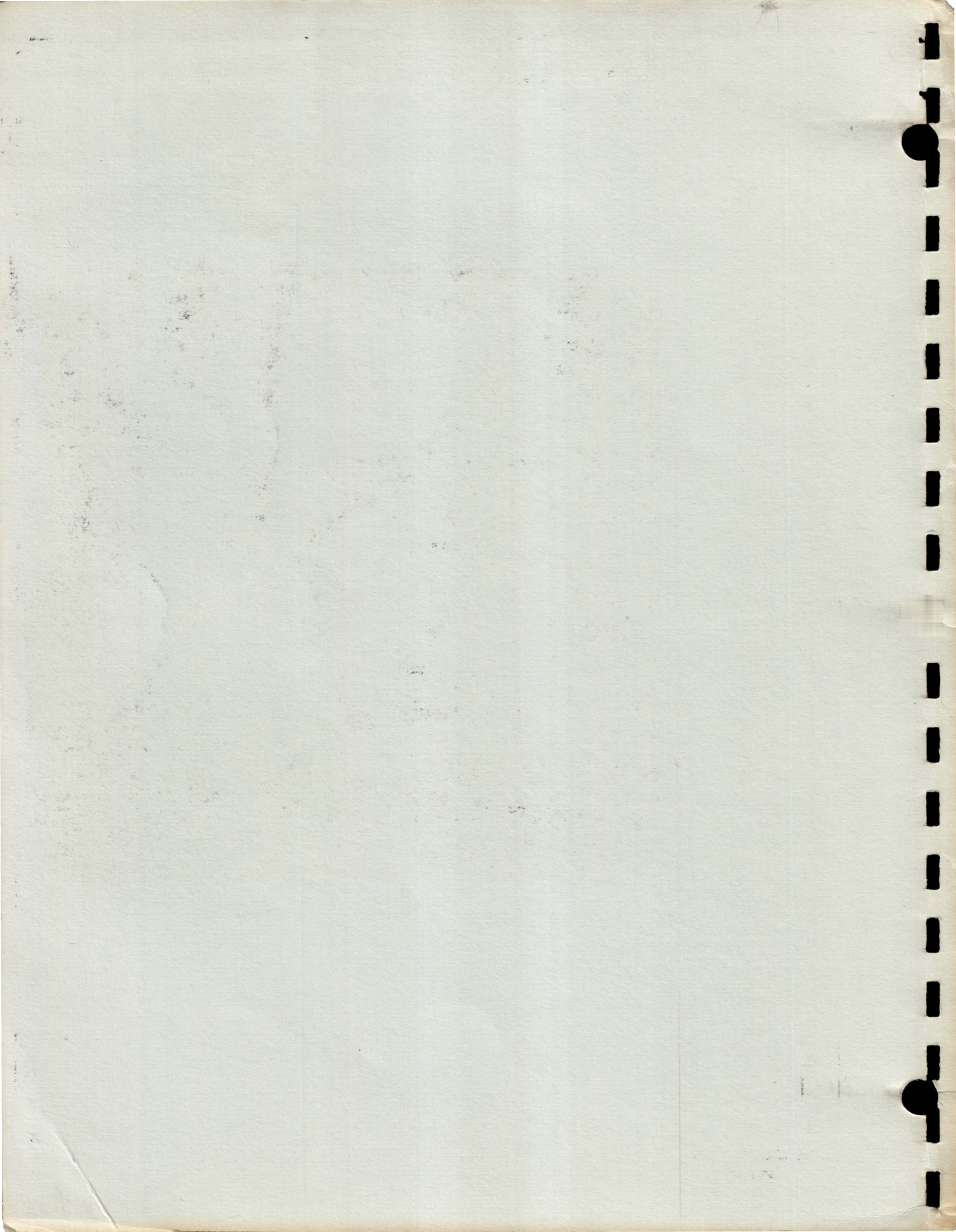
RELATIVE TO
III.Ø


B 7000/B 6000 Series

CANDE

REFERENCE MANUAL

PRICED ITEM



Burroughs 

B 7000/B 6000 Series

CANDE

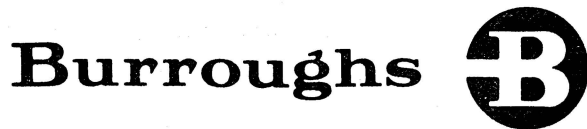
REFERENCE MANUAL

Copyright © 1980, Burroughs Corporation, Detroit, Michigan 48232

PRICED ITEM

Burroughs believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material. The information contained herein is subject to change. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this document should be addressed directly to Burroughs Corporation, P.O. Box 4040, El Monte, California 91734, Attn: Publications Department, TIO-West.



RDOK 10380

PUBLICATION
CHANGE ^{u6415}
NOTICE

PCN No.: 5011398-001 Date: March 1981
Publication Title: B 6000/B 7000 CANDE Language Reference Manual

Other Affected Publications: None

Supersedes: N/A

Description:

	<u>Change Pages</u>	<u>Add Pages</u>
TOC-1	4-81	3-2A
TOC-3	4-83	4-3B
TOC-5	4-89	4-19B
1-5	4-97	4-22A
3-1	4-103	
3-3	4-109	
3-9	5-1	
4-3	5-11	
4-19	5-25	
4-29	5-45	
4-49	6-1	
4-55	Index-1	
4-79	Index-3	

Retain this PCN as a record of changes made to the basic publication.

The above pages covering
PCN 5011398-001

COPYRIGHT © 1981
BURROUGHS CORPORATION
Detroit, Michigan 48232

Burroughs believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material. The information contained herein is subject to change. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this document should be addressed directly to Burroughs Corporation, P.O. Box 4040, El Monte, California 91734, Attn: Publications Department, TIO-West.

Burroughs



PUBLICATION
CHANGE
NOTICE

BOOK 12351

PCN No.: 5011398-002 Date: March 1, 1982

Publication Title: B 5000/B 6000/B 7000 Series CANDE Reference Manual (June 1980)

Other Affected Publications: _____

Supersedes: N/A

Description: Incorporates the Mark 3.3 System Software release.

Replace these pages

Add these pages

TOC-1	4-51	4-10-A
TOC-3	4-53	4-52-A
TOC-5	4-55	4-53-B
1-1	4-73	4-53-D
3-2A	4-95	4-95-B
4-9	4-115	5-22-A
4-31	4-117	
4-37	4-119	
4-39	Index-1	
4-43	Index-3	
4-45	Index-5	

CANDE REFERENCE MANUAL

TABLE OF CONTENTS

1. INTRODUCTION.	1-	1
RAILROAD DIAGRAMS	1-	2
RAILROAD COMPONENTS	1-	3
2. GENERAL INFORMATION	2-	1
USER IDENTIFICATION	2-	1
MULTIPLE COMMANDS	2-	1
RESPONSES	2-	1
SAVED TEXT QUEUE MANIPULATION	2-	1
INPUT QUEUE MANIPULATION.	2-	2
FILES	2-	3
Workfiles	2-	3
RECOVERY FILES.	2-	3
REMOTE FILES.	2-	5
3. BASIC CONSTRUCTS.	3-	1
<base>.	3-	1
<compiler name>	3-	1
<compiler type>	3-	1
<directory name>.	3-	1
<edit specs>.	3-	2
<end column>.	3-	2
<family name>	3-	2
<file identifier>	3-	2
<file name>	3-2-	A
<file title>.	3-2-	A
<host identifier>	3-2-	A
<identifier>.	3-	3
<inc>	3-	3
<integer>	3-	3
<mnemonic>.	3-	3
<modifier>.	3-	4

<name>	3- 5
<number>	3- 5
<sequence range>	3- 5
<sequence range list>	3- 5
<special char>	3- 6
<standard compiler>	3- 7
<start column>	3- 7
<string>	3- 7
<type>	3- 8
<usercode>	3- 8
4. CANDE COMMANDS	4- 1
ACCESS	4- 2
ADD	4- 3
APASSWORD	4- 4
BACKUPPROCESS	4- 6
BDREMOVE	4- 10
BIND	4- 11
BYE	4- 12
CHANGE	4- 13
CHARGE	4- 14
COMPILE	4- 15
CONTINUE	4- 17
COPY	4- 19
DCSTATUS	4- 20
DELETE	4- 22
DESTNAME	4-22- A
DISCARD	4- 23
DO	4- 24
EXCLUDE	4- 26
EXECUTE	4- 27
FAMILY	4- 28
FILES	4- 30
FIND	4- 31

FIX	4- 34
GET	4- 38
HELLO	4- 40
INSERT.	4- 41
LFILES.	4- 43
LIST.	4- 46
LOAD.	4- 49
LOG	4- 50
MAKE.	4- 51
MARGIN.	4- 53
MARKID.	4-53- A
MATCH	4- 54
MCS	4- 57
MERGE	4- 58
MOVE.	4- 59
NEWS.	4- 61
PASSWORD.	4- 62
PRINT	4- 64
RANGE	4- 65
RECOVER	4- 67
REMOVE.	4- 68
RENEW	4- 70
REPLACE	4- 71
RESEQ	4- 75
RMERGE.	4- 77
RO.	4- 78
RUN	4- 79
SAVE.	4- 80
SCHEDULE.	4- 82
SECURITY.	4- 86
SEQ	4- 88
Single-Line Entry or Deletion	4- 90
SO.	4- 93

SPLIT	4- 94
START	4- 95
STATUS	4- 96
STOP	4- 97
TAB	4- 98
TAPE	4-100
TERMINAL	4-102
TITLE	4-104
TYPE	4-107
UPDATE	4-108
UTILITY	4-110
WFL	4-114
WHAT	4-115
WRITE	4-116
5. CONTROL COMMANDS	5- 1
?AX	5- 2
?C	5- 3
?CHAR	5- 4
COMMENT	5- 5
?COUNTS	5- 6
?CS	5- 7
?CU	5- 8
?DC or ?ID	5- 9
?DENY	5- 10
?DP or ?DUMP	5- 11
?DS	5- 12
ECHO Function	5- 13
?EDIT	5- 14
?END	5- 15
?ENTER	5- 16
?EOL	5- 18
?FA	5- 19
?FR	5- 20

?GO	5- 21
?HI	5- 22
?HN	5-22- A
?JA	5- 23
?MCS.	5- 24
?MIXA or ?MXA	5- 25
?MM	5- 26
?MSG.	5- 27
?NUMBERED and ?UNNUMBERED	5- 28
?OF	5- 29
?OK	5- 30
?OT	5- 31
?PC or ?SC.	5- 32
?PURGE.	5- 33
?REPEAT	5- 34
?REPORT	5- 35
?RESTART and ?NORESTART	5- 36
?RESUME	5- 37
?RETRIEVE	5- 38
?RM	5- 39
?SCHEDULE	5- 40
?SHOW	5- 41
?SQ	5- 42
?SS	5- 43
?ST	5- 44
?STATUS	5- 45
?TAKE	5- 46
?TF	5- 47
?TI	5- 48
?TIME	5- 49
?TO	5- 50
?WAIT	5- 51
?WD or ?TD.	5- 52

?WHERE.	5- 53
?WHY.	5- 54
?WI or ?SI.	5- 55
?WM	5- 56
?WRU.	5- 57
?WS	5- 58
?WT	5- 59
?Y.	5- 60
6. FUNCTIONAL COMMAND GROUPINGS.	6- 1
Task Commands	6- 1
Library Commands.	6- 1
Workfile Commands	6- 1
Search Commands	6- 1
Edit Commands	6- 1
User Information.	6- 1
Utility Commands.	6- 1
Session Commands.	6- 1
System Inquiry Commands	6- 1
Task Interrogation and Control Commands	6- 2
Schedule Control Commands	6- 2
End-of-File Commands.	6- 2
Data Comm Network Commands.	6- 2
Input Editing	6- 2
Queue Manipulation.	6- 2

1. INTRODUCTION

The Command and Edit (CANDE) language for the B 7000/B 6000 series of computer systems provides generalized file preparation and updating capabilities in an interactive, terminal-oriented environment. Compilation and execution of programs with data input/output is provided by the CANDE Message Control System (MCS).

The CANDE manual is divided into six sections:

Section	Contents
1. INTRODUCTION	A brief introduction to CANDE and an explanation of railroad syntax is described in this section.
2. GENERAL INFORMATION	Generalized information regarding the capabilities and use of CANDE are explained in this section.
3. BASIC CONSTRUCTS	The basic constructs containing syntax diagrams of terms used throughout the manual are presented in this section.
4. CANDE COMMANDS	The CANDE commands used for such functions as task initiation, library manipulation, and workfile creation and editing are listed in this section.
5. CONTROL COMMANDS	The control commands that provide the ability to control and interrogate the operating environment are listed in this section.
6. FUNCTIONAL COMMAND GROUPINGS	Functional command groupings that illustrate how the different CANDE commands are associated are described in this section.

The following documents contain related information.

Publication Form No.

B 7000/B 6000 SERIES CANDE REFERENCE CARD	5001349
B 7000/B 6000 SERIES CANDE OPERATIONS MANUAL	5011562

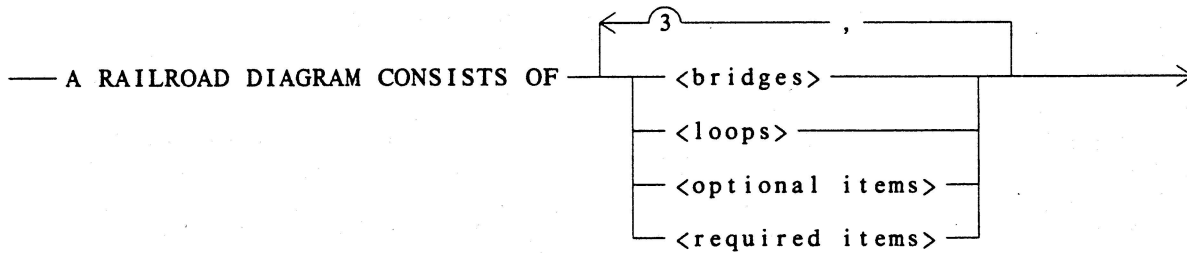
RAILROAD DIAGRAMS

Railroad diagrams are graphically oriented techniques used to represent the syntax of utility input statements, language verbs, and operator display terminal commands.

The railroad diagrams are traversed left-to-right or in the direction the arrowhead points. Adherence to the limits illustrated by bridges produces a syntactically valid statement. Continuation from one line of a diagram to the next line is represented by a right arrow (>) appearing at the end of the current line and at the beginning of the next line. The complete syntax diagram is terminated by a vertical bar (|) or a percent (%).

Items contained in broken brackets (<>) are syntactic variables that are defined in the manual or are information the user is required to supply.

The items in upper case must appear literally, and the minimum abbreviations are underlined.

Example

— AND IS TERMINATED BY A VERTICAL BAR OR PERCENT. —

The following syntactically valid statements are some statements that may be constructed from the above diagram:

A RAILROAD DIAGRAM CONSISTS OF <bridges> AND IS TERMINATED BY A VERTICAL BAR OR PERCENT.

A RAILROAD DIAGRAM CONSISTS OF <optional items> AND IS TERMINATED BY A VERTICAL BAR OR PERCENT.

A RAILROAD DIAGRAM CONSISTS OF <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR OR PERCENT.

A RAILROAD DIAGRAM CONSISTS OF <optional items>, <required items>, <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR OR PERCENT.

RAILROAD COMPONENTS**<required items>**

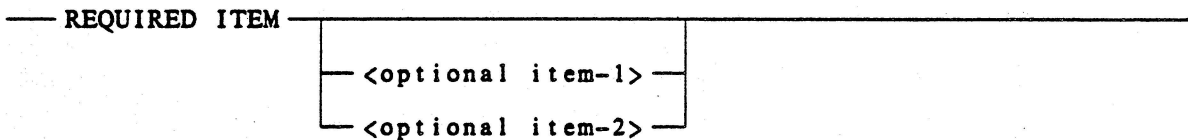
No alternate path through the railroad diagram exists for required items or required punctuation.

Example

— REQUIRED ITEM —

<optional items>

Items shown as a vertical list indicate that a choice of the items specified may be made. An empty path through the list allows the <optional item> to be absent.

Example

The following valid statements may be constructed from the above diagram:

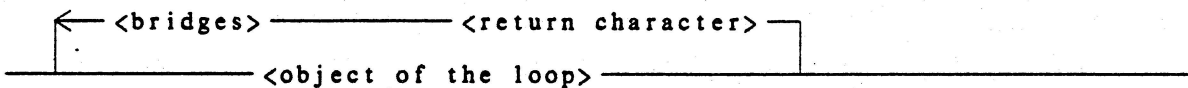
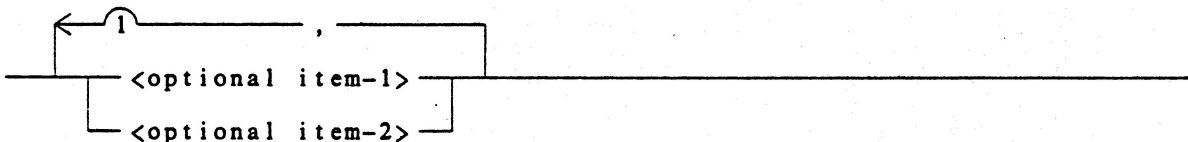
REQUIRED ITEM

REQUIRED ITEM <optional item-1>

REQUIRED ITEM <optional item-2>

<loops>

A <loop> is a recurrent path through a railroad diagram and has the following general format:

**Example**

The following statements can be constructed from the above diagram:

<optional item-1>

<optional item-1>,<optional item-1>

<optional item-2>,<optional item-1>

A <loop> must be traversed in the direction the arrowhead points, and the limits specified by bridges cannot be exceeded.

CANDE REFERENCE MANUAL

<bridge>s

traversed in a railroad diagram.

The following illustrations show the two forms of <bridges>.

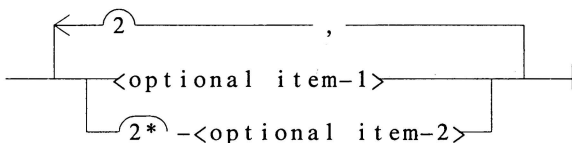
/ n

In this case, n is an integer that specifies the maximum number of times the path may be traversed.

/ n*

The above illustration shows n as an integer that specifies the minimum number of times the path must be traversed; the asterisk (*) indicates that the path must be traversed at least once.

Example



The loop may be traversed a maximum of two times. The path for <optional item-2> must be traversed at least one time, but no more than twice.

The following statements can be constructed from the above diagram:

<optional item-1>,<optional item-2>

<optional item-2>,<optional item-2>,<optional item-1>

<optional item-2>

CANDE REFERENCE MANUAL

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES. |

2. GENERAL INFORMATION

USER IDENTIFICATION

To use CANDE, the terminal must be active and logged-on the system. Also, a valid usercode and password must be presented to the system. The usercode is entered first, followed by the password. If the terminal used is a hardcopy device, the password may be typed into a blacked-out area. For screen devices, a similar degree of password security may be obtained by decreasing the screen illumination before entering the password. The usercode/password combination may be entered on the same line and separated by a slash (/) or a space.

CANDE checks the usercode/password against the file of authorized users (SYSTEM/USERDATAFILE) of the system. If the usercode/password is not valid, the user is informed and asked to re-enter the information. If the usercode/password is valid, CANDE responds with an appropriate message and the user is logged-on. CANDE may not be used until successful log-on has been achieved.

The usercode is used to identify files in the library of the user. The usercode is appended to the front of the file name. (For example, (MYUZER)DUMMYFILE ON PACK where MYUZER is the usercode.)

MULTIPLE COMMANDS

More than one CANDE command (control commands excluded) may be entered on a single line by separating the commands with semicolons (;). The commands ADD, BYE, COPY, FIX, HELLO, SEQ, TAPE, WFL, and Single Line Entry or Deletion, however, must appear alone on a line or as the last command of that line. This information is shown in the railroad diagram of each command; a bar (|) termination symbol indicates multiple entry is legal, and a percent (%) symbol denotes that this command must be the last or only command on that line.

RESPONSES

All commands input to CANDE result in the output of the desired information followed by a pound sign (#) or a single pound sign (#) indicating that no immediate response is necessary. Also, a pound sign followed by an error message or messages related to the command may be returned.

SAVED TEXT QUEUE MANIPULATION

Each user has a queue of "saved text" that CANDE allows for reference. The most recent saved text may be edited and processed as the next line of input. Throughout this manual, the use of saved text without a queue entry number refers to the most recent entry. Each normal line of input (a line not beginning with the control character defined for the user's station) becomes the current saved text. Each entry in the queue is then moved up one level. If the saved text queue contains the maximum number of entries, the oldest entry is discarded. The maximum number of entries that can be saved in the saved text queue can be altered by a CANDE control command administered by the operations manager.

CANDE REFERENCE MANUAL

If a line of input contains more than one CANDE command, each command is entered separately in the saved text queue. If a command does not complete successfully, the most recent saved text entry contains the failed command. Commands that follow the failed command are either part of this entry or are the first entry in the input queue (depending on the nature of the failure).

The commands provided for listing and editing of prior commands are: ?SHOW, ?EDIT, ?REPEAT, and ?RETRIEVE.

INPUT QUEUE MANIPULATION

The number of queued entries executable by CANDE is unlimited. Queued input is in one of three states: (1) normal, (2) pending, or (3) waiting.

1. The normal state occurs when CANDE is currently busy executing a command for a station and another command is entered prior to the completion of the first. If the currently executing command completes normally, then the first queued command becomes the currently executing command, and so on. If an error occurs on the currently executing command and queued input exists, the queued input is then either pending or waiting. (Refer to the QWAIT option below.)
2. If the pending state occurs, and a normal line of input is given in response to the #QUEUED INPUT PENDING message, the queued input is discarded and the session is continued with that line of input. If the queued input should not be discarded, one of the queue manipulation commands may be used, and the input queue may be set to the waiting state.
3. In the waiting state, any normal line of input is queued at the end of the queue, and the message is QUEUED. A QUEUED INPUT WAITING message is then displayed. The state of the queue input remains waiting until a ?GO, ?PURGE, or ?REPEAT command is issued.

While the state of the queued input is waiting or pending, the queued input entries may be entered, removed, or edited. Insertion of an entry may specify that CANDE is to wait for intervention before performing that entry, or the session may be resumed with the first entry in the queue or with the saved text.

Certain queue manipulation commands are limited to operation on the visible portion of the user's queue. Only the first 20 entries of the user's queue are visible.

QWAIT is a user option that may be set or reset with the SO and RO commands, respectively. If QWAIT is set and an error occurs, the state of the user's queued input is set to waiting; otherwise, it is set to pending.

The commands provided for manipulation of queued entries are: ?GO, ?WAIT ?PURGE, ?TAKE, and ?ENTER.

FILES

Files created through CANDE have the following attributes:

```
KIND = DISK
UNITS = WORDS (CHARACTERS for CDATA and CSEQDATA)
MAXRECSIZE = 14, 15, or 80 (see Table 3-1 in Section 3.)
MINRECSIZE = 0
BLOCKSIZE = 420 (2160 for CDATA and CSEQDATA)
AREAS = 15
AREASIZE = 504 (1134 for CDATA and CSEQDATA)
SAVEFACTOR = 30
SECURITYUSE = IO
SECURITYTYPE = PRIVATE
```

Workfiles

CANDE can access any file in the user's library for various purposes, but changes may only be effected on the workfile. A new workfile may be made using the MAKE command, or an existing file may be recalled as the workfile using the GET command. All editing commands apply to the workfile. Single-line entries, as well as FIX commands, are not applied to the workfile immediately but are tanked until the next UPDATE.

RECOVERY FILES

The workfile currently being updated resides in a TEXT file separate from the compiled object workfile that exists as a CODE file. All other information about an active workfile, including any changes since the last update, is kept in the tankfile. A recovery file is created by transcribing the information in the tankfile which pertains to the station whose session was aborted. This transcription is performed immediately when CANDE is next initiated.

The workfile title for the recovery file is CANDE/TEXT<recovery number>; the workfile object code has the title of CANDE/CODE<recovery number>. The recovery number for a session is determined at the beginning of the session by using the system; this number is suffixed to the TEXT and CODE files created by updating and compiling the workfile. If a recovery file must be created, the same number is suffixed to the RECV file. The recovery number consists of the logical station number (in decimal) followed by a digit to distinguish among multiple recovery files from the same station.

This scheme imposes a limit of 10 recovery files from the same station, and a total of 25 recovery files under any one usercode.

A high level of consistency checking is applied to recovery files to screen out files harmful to CANDE. When an invalid recovery file is detected, the following actions are taken: first, the contents of the RECV file are listed in the CANDE taskfile to permit diagnosis of the failure. The file is then purged. If a TEXT workfile exists (that is, if the file was updated since GET or MAKE), then that file is recovered. The results of such recovery are as follows:

```
#WORKFILE IS NOT NAMED:  ALGOL, 347 RECORDS
```

```
INVALID RECOVERY FILE; NAME AND ANY CHANGES WERE LOST
```

CANDE REFERENCE MANUAL

If no text existed or if the recovery failed, the message is:

```
#INVALID RECOVERY FILE
```

If a CANDE fault or error occurs in a workfile editing or output operation (a CANDE "worker"), the following actions are taken to invoke the consistency checking of CANDE workfile recovery:

1. The tankfile data are saved in a recovery file, as though the station had disconnected.
2. CANDE displays #AUTORECOVERY INITIATED.
3. The action of a RECOVER command is taken using the appropriate recovery file.

If the consistency checking fails, normal INVALID RECOVERY FILE action follows.

Recovery information is contained in one of three files:

1. A RECOVERY file containing any workfile changes since the last update, as well as the title and other attributes of that workfile.
2. A TEXT file is created if the workfile has been updated but not yet saved.
3. A CODE file is generated if the workfile has been compiled but not yet saved.

A TEXT or CODE file is generated at update or compilation time and is written into the workfile family. The recovery file is produced when or after the session is aborted and may be written on one of two families. If possible, the recovery file is written on the DEFAULT WORKFILE FAMILY, as defined by the workfile family specifications established by the user at log-on time. If that family is not available, the recovery file is written on the family containing the codefile for the CANDE MCS.

The RECOVER command displays and recovers recovery files in up to three places in the following order of precedence: (1) the userdata default workfile family, (2) the CANDE MCS family, and (3) the current workfile family. The DISCARD command removes recovery, TEXT, and CODE files with the specified number(s) from all three of these families.

A recovery-data display is grouped according to the family containing the recovery files. If a recovery file pertains to a workfile on a different family, the phrase ON <family name> appears in the display.

After recovering a workfile, the family specifications for that session are set to those family specifications in effect at the time the workfile was saved. The new specifications are displayed if they differ from those in effect before recovery.

If the file part of a recovered workfile is not present, the recovery action is aborted with an appropriate message. However, the recovery file is not purged and a recovery may then be attempted by making the missing file present.

CANDE REFERENCE MANUAL

REMOTE FILES

Remote files are of a special nature, in that an object program may treat a remote terminal as a file by setting the KIND attribute to REMOTE.

More than one open file may be assigned simultaneously to a terminal. However, more than one input or input/output file may never be open at the same terminal at the same time.

When a remote file is opened input or input/output, all input received from the terminal is considered input to the file and not a command to CANDE. The only exception occurs when a line is preceded by the control character of the station. In that case, the input is to be a control command and is sent to CANDE for processing.

Output may be directed to another CANDE terminal by specifying the <lsn> of the terminal via the STATION modifier. For example, E;STATION=15 directs the remote output to <lsn> 15. Remote files may also be assigned to other CANDE terminals through use of the TITLE attribute in conjunction with setting the STATION modifier to zero. For example, E;STATION=0;FILE R(TITLE=TTY3) directs file R to the station named TTY3.

3. BASIC CONSTRUCTS

Several basic constructs are common to several CANDE commands. These constructs are defined in this section.

<base>

<base> is an integer.

<compiler name>

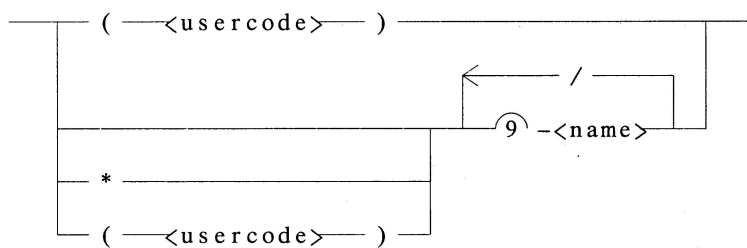
<compiler name> is a <file title>.

<compiler type>

<compiler type> is a <type>.

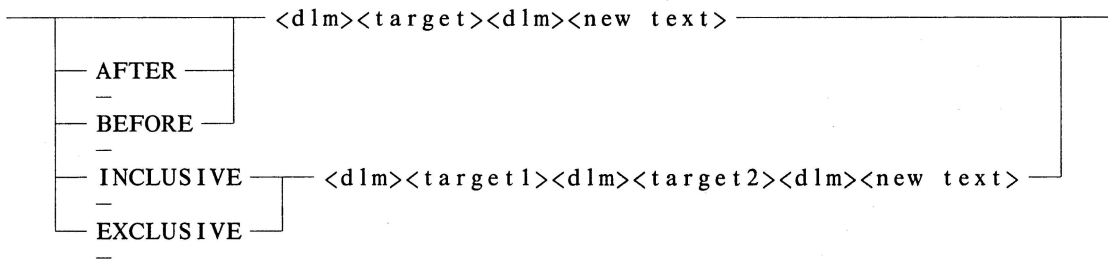
<directory name>

The following syntax defines the <directory name>:



<edit specs>

The syntax unit <edit specs> is used in several commands and is semantically and syntactically a subset of the FIX command. The following syntax defines the <edit specs>:



<end column>

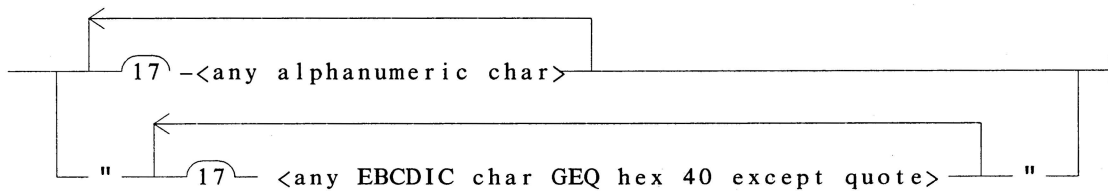
<end column> is an integer.

<family name>

<family name> is an identifier.

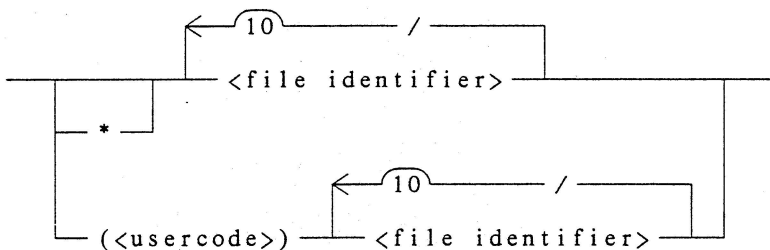
<file identifier>

The following syntax defines the <file identifier>:



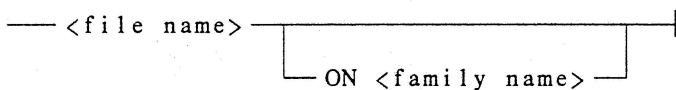
<file name>

The following syntax defines the <file name>:



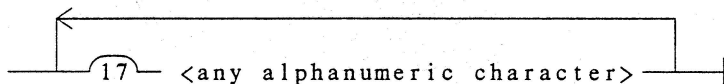
<file title>

The following syntax defines the <file title>. If a <file title> consists of an <integer>, it may be enclosed in quotation marks to avoid confusion with a <sequence number>.



<host identifier>

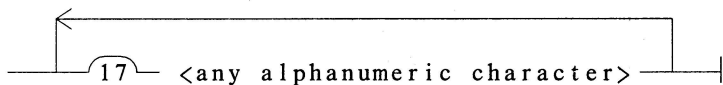
The following syntax defines the <host identifier>:



THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES.

<identifier>

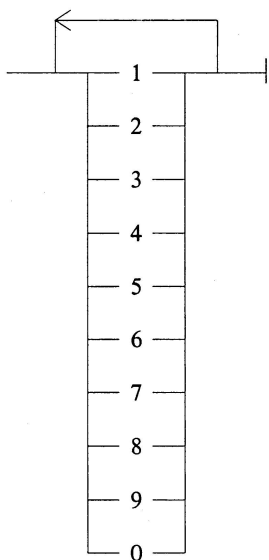
The following syntax defines the <identifier>:

**<inc>**

<inc> is an integer.

<integer>

The following syntax defines <integer>:

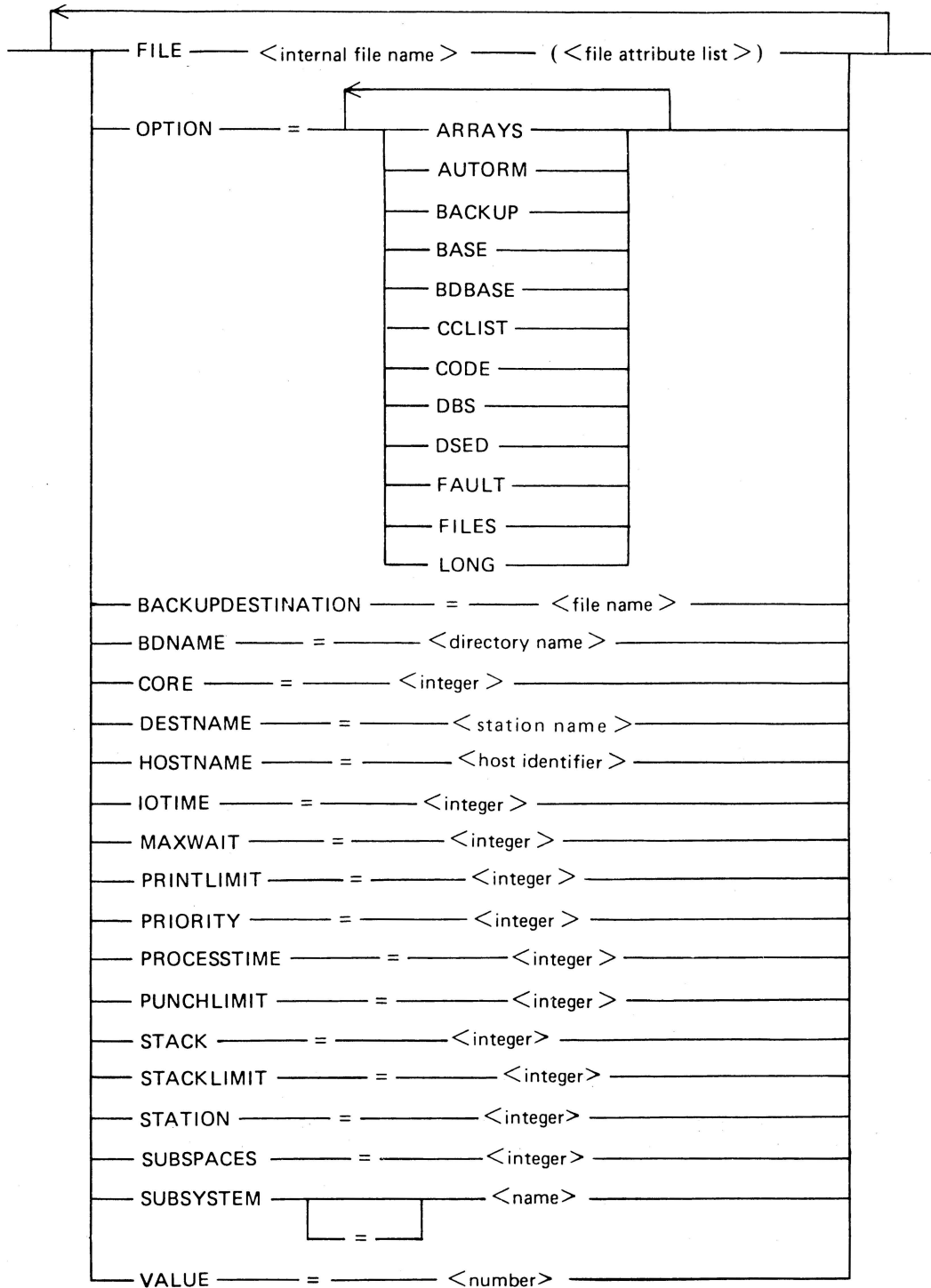
**<mnemonic>**

The following list defines <mnemonic>:

NUL SOH STX ETX HT DEL VT FF CR SO SI DLE DC1 DC2 DC3 NL
 BS CAN EM FS GS RS US LF ETB ESC ENQ ACK BEL SYN EOT NAK
 SUB SP

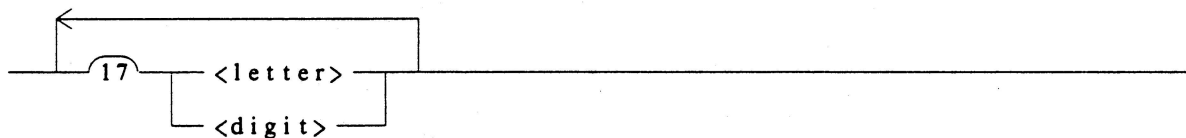
<modifier>

The following syntax defines <modifier>:

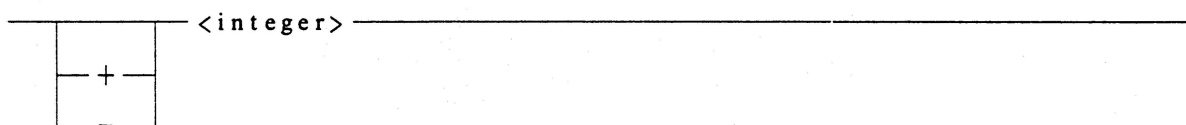


<name>

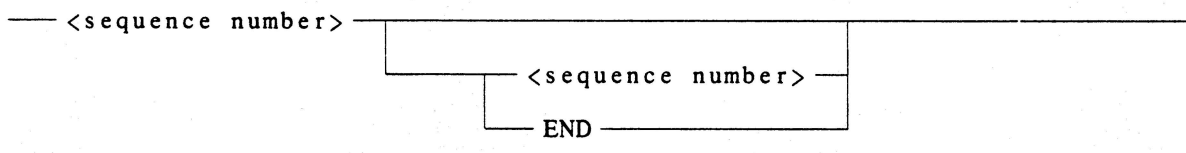
The following syntax defines <name>:

**<number>**

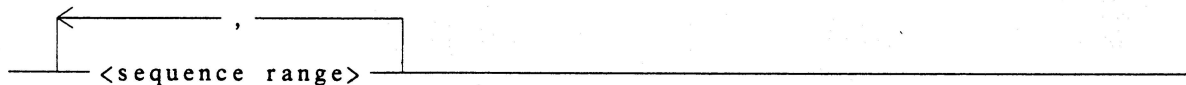
The following syntax defines <number>:

**<sequence range>**

The following syntax defines <sequence range>:

**<sequence range list>**

The following syntax defines <sequence range list>:



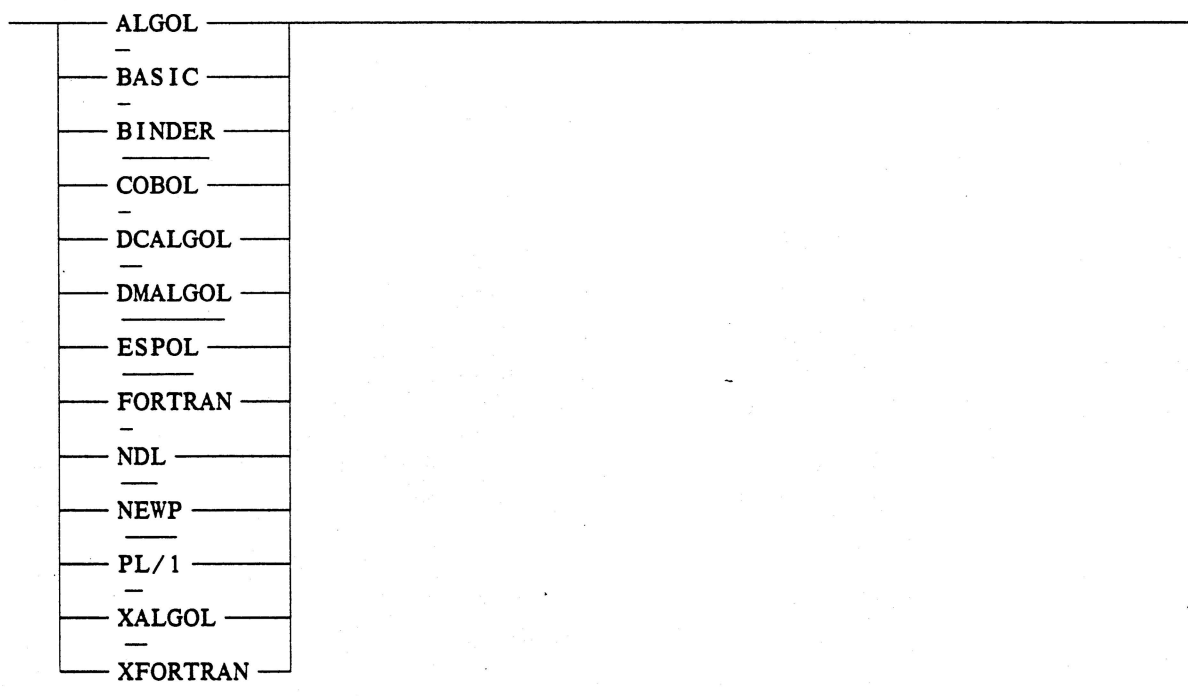
<special char>

The following syntax defines <special char> but is not a complete list of special characters.

"
#
\$
%
&
'
(
)
=
-
@
<
>
,
.
/

<standard compiler>

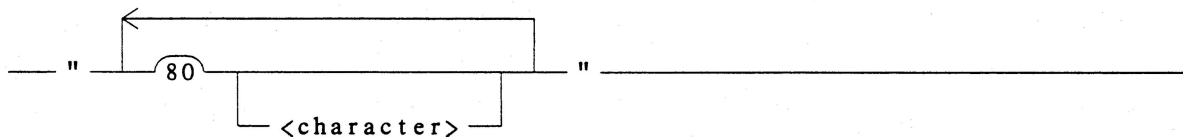
The following syntax defines <standard compiler>:

**<start column>**

<start column> is an integer.

<string>

The following syntax defines <string>:



A pair of quote characters appearing alone represents a null string.

A pair of quote characters appearing within a string represents a single quote character in the string.

<type>

The following syntax defines <type>:

ALGOL
-
COBOL
-
BASIC
-
FORTRAN
-
PLI
-
XALGOL
-
NDL
-
DCP
-
XFORTRAN
-
BINDER
-
DASDL
-
GUARD
-
DCALGOL
-
DMALGOL
-
ESPOL
-
SEQ
-
DATA
-
CDATA
-
JOB
-
NEWP
-

<usercode>

<usercode> is a <file identifier>.

Table 3-1 summarizes the record formats used for each CANDE type.

Table 3-1. Record Formats

Type	Text Field	Sequence Field	ID Field	Record-length		Compiler Type
				Default	Minimum	
ALGOL	1-72	73-80	81-90	15 WD	80 CH	YES
BASIC	5-72	1-4	73-80	14 WD	72 CH	YES
BINDER	1-72	73-80	NA	14 WD	80 CH	YES
CDATA	1-80****	NA	NA	80 CH	NA	NO
COBOL	7-72	1-6	73-80	14 WD	72 CH	YES
CSEQ	7-80****	1-5	NA	80 CH	NA	NO
DASDL	1-72	73-80	81-90	15 WD	80 CH	YES
DATA	1-80**	NA	NA	14 WD	80 CH	NO
DCALGOL	1-72	73-80	81-90	15 WD	80 CH	YES
DCP	1-72	73-80	81-90	15 WD	80 CH	NO
DMALGOL	1-72	73-80	81-90	15 WD	80 CH	YES
ESPOL	1-72	73-80	81-88	15 WD	80 CH	YES
FORTRAN	1-72	73-80	NA	14 WD	80 CH	YES
GUARD*	1-72	73-80	NA	14 WD	80 CH	NO
JOB	1-80	83-90	NA	15 WD	90 CH	NO
NDL	1-72	73-80	81-90	15 WD	80 CH	YES
NEWP	1-72	73-80	81-88	15 WD	80 CH	YES
PL-1	1-72	73-80	81-90	15 WD	80 CH	YES
SCHEDULE	1-80*****	NA	NA	80 CH	NA	NO
SEQ	1-72	73-80	81-90***	14 WD	80 CH	NO
			TEXT DELETED			
XFORTRAN	1-72	73-80	NA	14 WD	80 CH	YES

CANDE REFERENCE MANUAL

NOTES

*A file of type GUARD is the output file of GUARDFILE and cannot be generated through CANDE.

**DATA file may have arbitrary length. The text field extends through the end of the record with one exception: by convention, 14-word EBCDIC or ASCII records are assumed to have text only through column 80.

***SEQ files are created with 14 words and no ID field. If CANDE encounters a SEQ file with 15 or more words (90 or more characters), columns 81-90 are treated as an ID field. (For example, such a file can be generated by making a file type ALGOL and changing its type to SEQ.)

****CDATA and CSEQ files have arbitrary length; the text field extends through the end of the record.

*****A SCHEDULE file may be of any type recognized by CANDE. (No type SCHEDULE exists.) The output for a SCHEDULE file is type SCHED. Any CANDE command treats a SCHED file as type CDATA.

4. CANDE COMMANDS

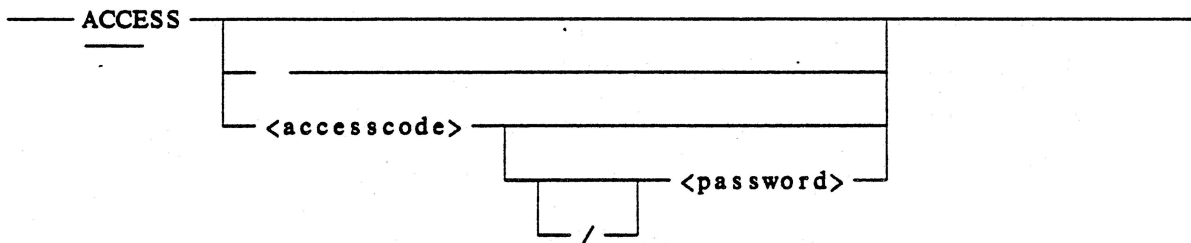
This section contains the CANDE commands for task initiation, workfile creation, user library manipulation, text editing, and so forth. The commands are presented in alphabetical order along with the railroad syntax structure, semantics, and simple examples.

Commands terminating with percent signs (%) must appear alone on the command line or as the last command of that line. Those commands ending with vertical bars may be grouped on a line with each command separated by a semicolon.

Section 6, FUNCTIONAL COMMAND GROUPINGS, provides a quick reference to related commands.

ACCESS

Syntax



Semantics

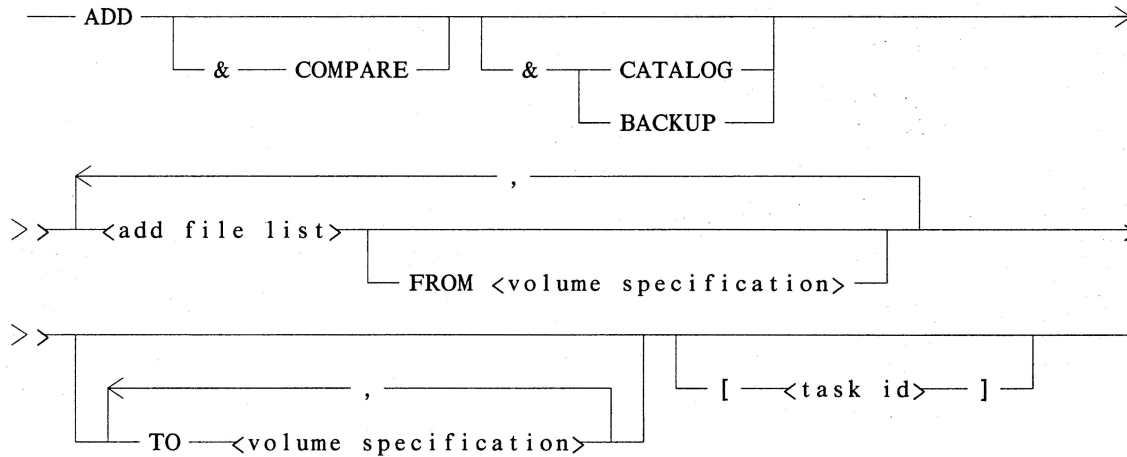
The ACCESS command is used to interrogate or change the ACCESSCODE of a session.

ACCESS returns the current accesscode for the session (not including the password).

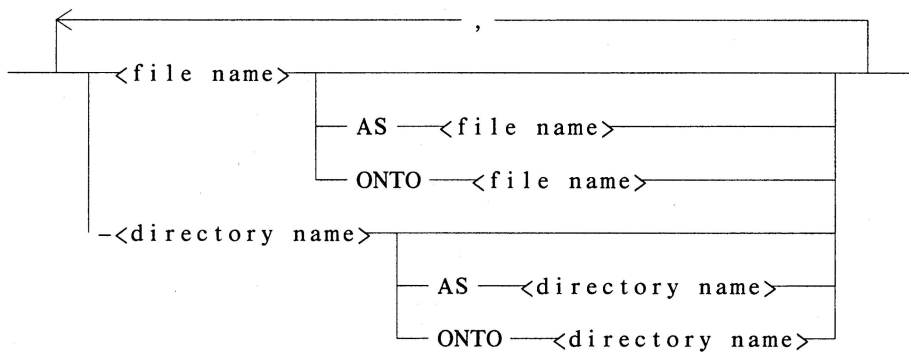
ACCESS . assigns a null accesscode to the session.

ACCESS <accesscode>/<password> assigns or changes the session accesscode after validation in the USERDATAFILE.

Changes of accesscodes to a session are logged.

ADD**Syntax**

<add file list>

**Semantics**

CANDE recognizes ADD to be a Work Flow Language (WFL) command and passes it and the specifications of the command to the WFL compiler for processing.

The ADD command is used to copy files to and from disks and tapes when the destination medium does not already contain the title of the file. That is, if a file is to be added to a destination where a file resides with the same title, then that file is not copied to the destination. For more detailed semantical information and the complete breakdown of syntax, refer to the ADD statement in Section 6 of the B 5000/B 6000/B 7000 Series Work Flow Language Reference Manual, form 5011794.

The ADD command is invalid for use with files in another user's directory.

Examples

```
ADD & COMPARE = FROM MYTAPE
#RUNNING 3042
#BOT 3043 (UZER)WFLCODE
#BOT 3044 LIBRARY/MAINTENANCE
#3044 (UZER)FILE1 ADDED.
#3044 (UZER)FILE2 ADDED.
#EOT 3044 LIBRARY/MAINTENANCE
#EOT 3043 (UZER)WFLCODE
#
```

```
ADD X/= FROM UZER(PACK) TO DISK
#RUNNING 3056
#BOT 3057 (UZER)WFLCODE
#BOT 3058 LIBRARY/MAINTENANCE
#3058 (UZER)X/B ADDED.
#3058 (UZER)X/C ADDED.
#3058 (UZER)X/D ADDED.
#EOT 3058 LIBRARY/MAINTENANCE
#EOT 3057 (UZER)WFLCODE
#
```

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES. |

CANDE REFERENCE MANUAL

The recovery file is not recovered or purged. If the recovery file has an accesscode, the accesscode is indicated in the list of recovery files as follows:

- # Accesscode is different from the accesscode of the session.
- * Accesscode is the same as the accesscode of the session.

CANDE recognizes the ACCESSCODE task attribute. For example, assume a session is running under usercode UA with accesscode AA/PA. The following is entered:

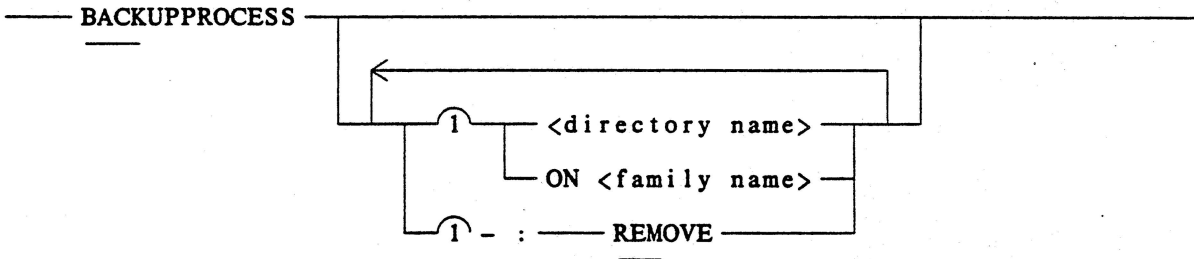
```
RUN (UB)P ON HISPACK; AC AB/PB
```

AB is the accesscode for the task (UB)OBJECT/P. At task start-up, AB/PB must be a valid accesscode/password for user UA; if not, the task is not initiated. Access to the program file (UB)OBJECT/P is determined by the security of the program code file; if the security is GUARDED or CONTROLLED, access is determined by the usercode UA and the accesscode AA of the session. Supplying an ACCESSCODE task attribute on a task does not change the accesscode of the session. The ACCESSCODE task attribute cannot be provided as a task attribute to a program being compiled from CANDE. For example, "C; C AC=A/B" is valid; however, "C; AC=A/B" is not valid.

Accesscodes are only required on tasks that access files which are protected by a guardfile which uses accesscodes to control access rights.

Any CANDE SCHEDULE sessions inherit the accesscode of the CANDE session. The ACCESS and APASSWORD commands are not valid within a CANDE SCHEDULE session. If a SCHEDULE session contains an ACCESS command with an invalid accesscode, the SCHEDULE session is aborted.

CANDE includes the accesscode in all log records about a session, particularly in log-on and log-off records.

BACKUPPROCESS**Syntax****Semantics**

The BACKUPPROCESS command allows backup files to be copied to disk, identified, removed, and listed. Backup files created in a session on DISK and PACK under the *BD and *BP directories, created in a session on any arbitrary disk family, or contained in a directory may be accessed.

Each backup file created in the current session is presented to the user. Proper responses to a presented backup file are NEXT, REMOVE, WHAT, COPY and LIST.

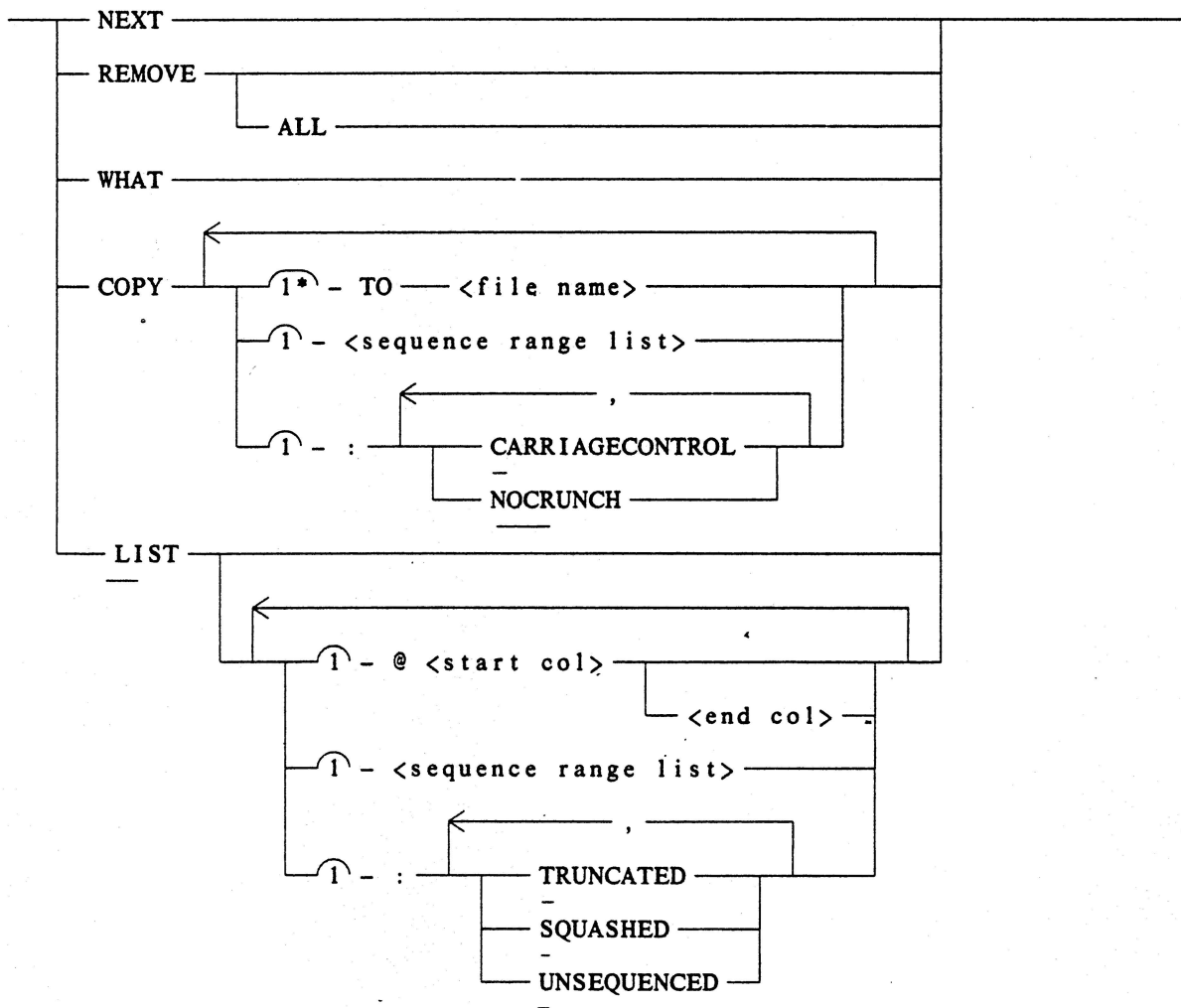
The CANDE BACKUPPROCESS command is implemented as a separate task and communicates with the user via a remote file. Termination occurs (1) when all backup files have been presented; (2) when an end-of-file indication for the remote file is detected; or (3) when the BACKUPPROCESS task is DSed.

If REMOVE is specified, all backup files found are removed without being presented.

<directory name> specifies the directory in the user's library that is to be searched for backup files. Only backup files found in this directory are presented to the user. The directory may only designate files in the user's library.

ON <family name> allows specification of an arbitrary disk family that is to be searched for *BD and *BP files created in the session.

Syntax for BACKUPPROCESS Command Responses



Semantics

NEXT

Entering NEXT in response to a presented backup file causes the next file to be presented.

REMOVE

Entering REMOVE in response to a presented backup file causes the presented file to be removed. If ALL is specified, the remaining files are removed without being presented; otherwise, the next file is presented.

CANDE REFERENCE MANUAL

COPY

The COPY response is used to create a disk file copy of the presented backup file. This disk file is created in the user's directory on the primary disk family of the user with the name <file name>. An error is issued, and the COPY is not performed if <file name> is in use by an existing file.

The disk file created has a MAXRECSIZE of 30 words, a BLOCKSIZE of 270 words, and an AREASIZE of 504 words. Each record from the backup file is translated to EBCDIC, if necessary, and padded on the right by blanks to fit a 30-word record.

An entry in the backup file with SPACE 2 carriage control is treated as two records. If the entry contains no text, two blank records are inserted in the disk file. If the entry contains text, one blank record is inserted in the disk file. All other entries are counted as one record. If the entries do not contain text, a blank record is inserted in the disk file.

The CARRIAGECONTROL option causes carriage control information for backup disk (BD) files to be encoded in each disk file record in CTL360 format. Use of this format means that the first character of each record contains carriage control information with data beginning in the second column. Because blank records are inserted as previously described, only SKIP and SPACE 0 carriage control information is encoded. The filekind of the disk file is BDDATA (value = 191) to allow easy external recognition. CANDE WRITER uses the encoded carriage control of BDDATA files and does not reformat them, unless a column range is specified which does not contain column 1. With this option, the representation of the BD file, or portions thereof, may be easily recreated if desired. If the file is a backup printer (BP) file, the CARRIAGECONTROL option is ignored.

If the CARRIAGECONTROL option is not used or the file is a BP file, the first character of each disk file record is the first character of data and the filekind of the disk file is DATA.

The NOCRUNCH option causes the disk file not to be crunched. If this option is not specified, the disk file is crunched.

The meaning of <sequence range list> for a backup file is analogous to the meaning of <sequence range list> for a disk file whose filekind is DATA. That is, CANDE assigns a sequence number equal to the record number multiplied by 100 (for example, 100,200,300 and so forth).

NOTE

Sequence ranges are not allowed with old backup files.

The same backup file is represented after it is copied.

WHAT

Entering WHAT in response to a presented backup file causes the title of the presented backup file and the name of the job that created it to be displayed. If the USERCODE and/or CHARGECODE of the job that created the backup file are other than those of the current session, they are also displayed. If the information is available, the number of records contained in the backup file is displayed.

LIST

Entering the LIST response causes the presented backup file, or portions thereof, to be listed at the user's terminal, translating each nongraphic character to an EBCDIC question mark (?).

An entry in the backup file with SPACE 2 carriage control is counted as two records. If the entry contains no text, two blank records are listed. If the entry contains text, one blank record is listed. All other entries are counted as one record and listed as such. Entries that do not contain text are listed as blank records. Each record listed is preceded by a sequence number (unless the :UNSEQUENCED option is used), as if a data file were being listed with the normal CANDE LIST command.

NOTE

Sequence ranges are not allowed with old backup files.

This scheme of record counting and sequence numbering means that a disk file created by the COPY option and listed later with a normal CANDE LIST command appears exactly as the original backup file would appear when listed with the BACKUPPROCESS LIST option.

The column range specifications and the TRUNCATED, SQUASHED, and UNSEQUENCED options function exactly as for the normal CANDE LIST command. The <sequence range list> is as described for the COPY option.

The same backup file is represented after it is listed.

Examples

```
BACK
#RUNNING 3177
#?
#ENTER 'NEXT', 'LIST', 'WHAT', 'REMOVE' OR 'COPY'
#FILE BD/0003003/0003142/000LINE ON PACK; 6324 RECORDS
NEXT
#FILE BD/0003003/0003147/000OUTFILE ON PACK; 4504 RECORDS
WHAT
#FILE BD/0003003/0003147/000OUTFILE ON PACK; 4504 RECORDS;
  LABEL=OMITTED; JOB 3003: "CANDE WRITER"
COPY TO SHOWBACK 1000-10000:NOCR
#FILE BD/0003003/0003147/000OUTFILE ON PACK;4504 RECORDS
REM
#3177 BD/0003003/003147/000OUTFILE REMOVED ON PACK PK080.
FILE BD/0003003/0003158/000OUTFILE ON PACK; 9 RECORDS
L @ 25-30 600-800
600 ILETYP
700 TERS);
800 <80("1
REM ALL
#REMAINDER OF FILES REMOVED
#3177 BD/0003003/003158/000OUTFILE REMOVED ON PACK PK080.
#3177 BD/0003003/003159/000OUTFILE REMOVED ON PACK PK080.
#
```

BDREMOVE**Syntax**

```
— BDREMOVE —|
```

Semantics

The BDRMOVE command allows backup printer and punch files generated during a session to be removed. The command fires up a program that presents the name of backup files for the session. Three options are then available:

1. Entry of a blank line, which leaves the presented backup file untouched.
2. Entry of REM, which causes the presented backup file to be removed.
3. Entry of ALL, which causes the presented backup file, as well as those backup files yet to be presented, to be removed.

File names for backup printer files are of the form:

```
BD/<7-digit job num>/<7-digit task num>/<3-digit index><int filename>
```

File names for backup punch files are of the form:

```
BP/<7-digit job num>/<7-digit task num>/<3-digit index><int filename>
```

NOTE

The BDRMOVE command will be deimplemented on the 3.4 release; the BACKUPPROCESS should be used to remove files. (Refer to BACKUPPROCESS in this section.)

Examples

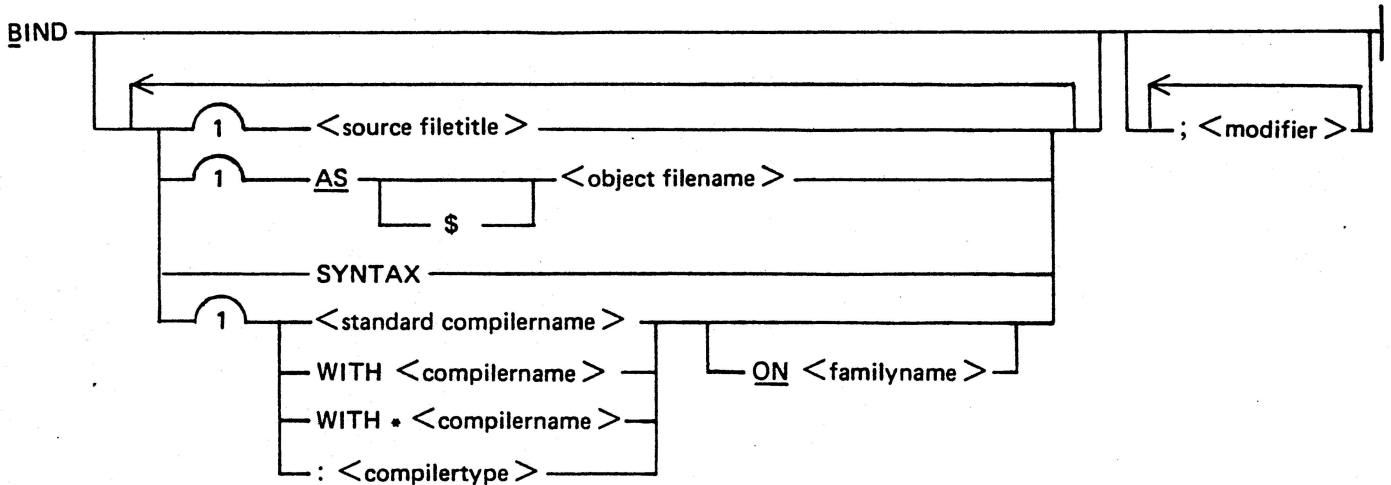
```
so msg
#MESSAGES SET
bdr
#RUNNING 3852
#?
#ENTER 'ALL' TO REMOVE REMAINDER OF FILES OR 'REM' TO REMOVE THIS
FILE
#SEND NULL INPUT TO SKIP THIS FILE
#FILE IS:BD/0003840/0003842/000OUTFILE ON PACK.
<this is a null input line>
#FILE IS:BD/0003840/0003843/000OUTFILE ON PACK.
```

CANDE REFERENCE MANUAL

4-10- A

rem
#3852 BD/0003840/0003843/000OUTFILE REMOVED ON PACK PK099 .
#FILE IS:BD/0003840/0003845/000OUTFILE ON PACK.
all
#REMAINDER OF FILES REMOVED
#3852 BD/0003840/0003845/000OUTFILE REMOVED ON PACK PK099 .
#3852 BD/0003840/0003847/000OUTFILE REMOVED ON PACK PK099 .
#

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES. |

BIND**Syntax**

MV2111

Semantics

The BIND command is similar to the COMPILE command except that if a compiler is not specified, the BINDER compiler is invoked.

Examples

```

BIND
#BINDING 3291
#ET=1:13.9 PT=6.3 IO=60.9
BIND ALGOL/PROG WITH ALGOL
#BINDING 3314
#ET=1:19.6 PT=8.9 IO=50.6
  
```

BYE**Syntax**BYE _____ %**Semantics**

The BYE command terminates the current CANDE session and disconnects dial-up lines.

If an unsaved workfile exists at the time BYE is entered, CANDE displays:

```
#REMOVE OR SAVE WORKFILE
```

This query indicates the presence of the unsaved workfile. The workfile must be removed or saved, and the BYE command re-entered, before the session can be terminated.

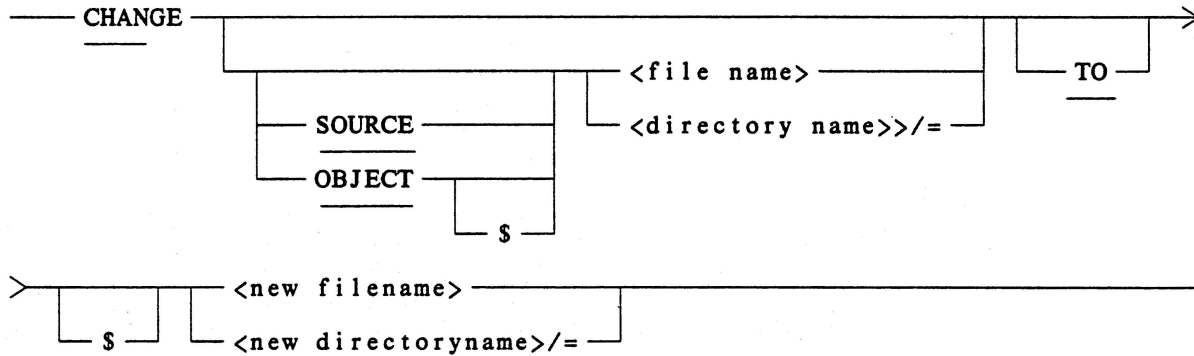
System resource usage statistics are provided on termination of the session. The session number, elapsed time of session (ET), processor time consumed (PT), I/O time consumed (IO), usercode, current time, and the date are printed in the following format:

```
#END SESSION <session number> ET=<time> PT=<time> IO=<time>
#USER = <usercode> <time> <date>
```

If a charge code is used for the session, then the charge code is printed following the usercode identification.

Example

```
BYE
#
#END SESSION 3003 ET=1:18:19.6 PT=1:08.9 IO=1:02.0
#USER = UZER 11:27:06 04/20/78
```

CHANGE**Syntax**

The CHANGE and TITLE commands are synonyms. (Refer to the TITLE command in this section for details.)

CHARGE

Syntax

```

CHARGE _____
  |
  |_____|
  |<charge code>|
  |_____|

```

Semantics

The CHARGE command is used to specify a new charge code at some point in the session. <charge code> must be an alphanumeric string not greater than 17 characters in length. If <charge code> is not specified, the charge code reverts to a no charge code specification. A new charge code can be entered as desired, thus providing a breakdown of charges at any point in the session. Entering a new charge code causes the system usage statistics for the session to be set to zero as if the previous session had terminated and a new one had begun.

Examples

```

charge
#SESSION 3601 ET=9.0 PT=0.0 IO=0.0
#CONTINUE SESSION 3601 14:08:03 4/20/78

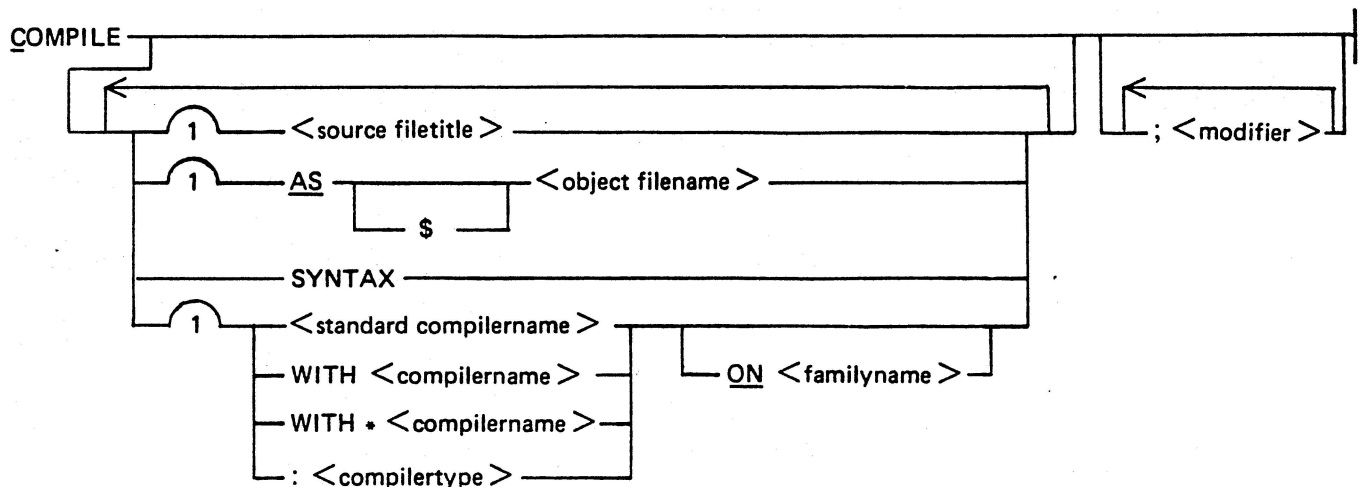
CHAR ME
#SESSION 3601 ET=7:47.2 PT=55.8 IO=57.1
#CONTINUE SESSION 3601 14:15:50 04/20/78

CHAR 12345678901234567
#SESSION 3601 ET=5:08.9 PT=0.5 IO=0.2
#CONTINUE SESSION 3601 14:20:59 04/20/78

```

COMPILE

Syntax



MV2112

Semantics

The COMPILE command is used to invoke a compiler. If <source filetype> is not specified, the workfile is assumed. The resulting object program is titled OBJECT/<object filename>, OBJECT/<source filetype>, or OBJECT/<workfile name>. If a dollar sign (\$) precedes the <object filename>, the OBJECT directory is omitted. The use of an asterisk (*) or <usercode> is not permitted with <object filename>.

If a compiler is not specified, <standard compiler> is invoked depending on the type of the source file. For example, the ALGOL compiler is invoked for type ALGOL files. If a compiler is not specified and the type of the source file (or workfile) is not a compiler type (such as SEQ or DATA), CANDE displays one of these messages:

```
#SOURCE NOT COMPILER TYPE
#WORKFILE NOT COMPILER TYPE
```

Nonstandard compilers (such as BDMSCOBOL and BDMSALGOL) may be invoked by using the WITH construct.

CANDE REFERENCE MANUAL

The SYNTAX option informs the compiler to compile for syntax only. No object file results if this option is specified.

Examples

```
c
#COMPILING 3673
#ET=7.2 PT=0.4 IO=0.4
```

```
COMPILE WITH BDMSCOBOL
#COMPILING 3698
#3721 BOJ *DATABASE/INTERFACE ON USERPACK
#3721/3721 EOJ JOB DATABASE/INTERFACE ON USERPACK
#ET=2:01.1 PT=1:03.8 IO=12.9
```

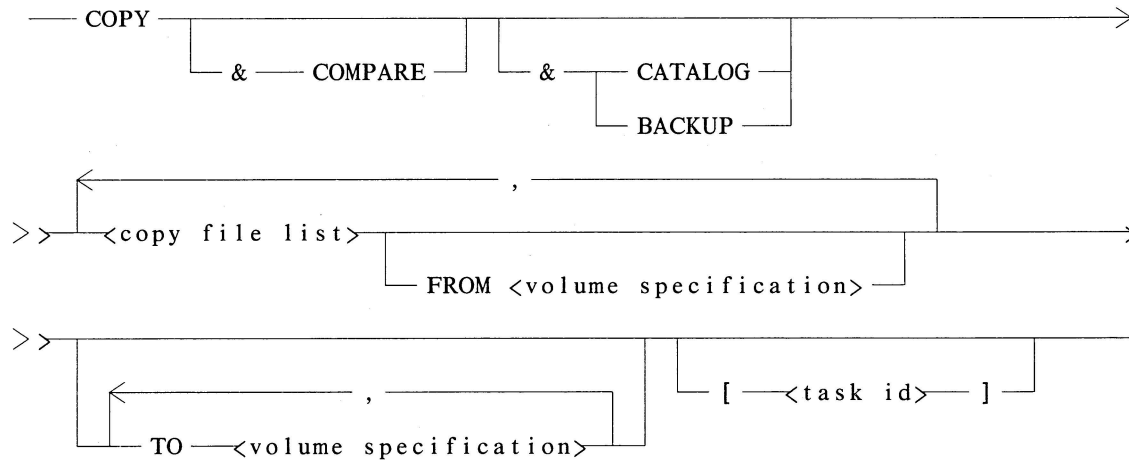
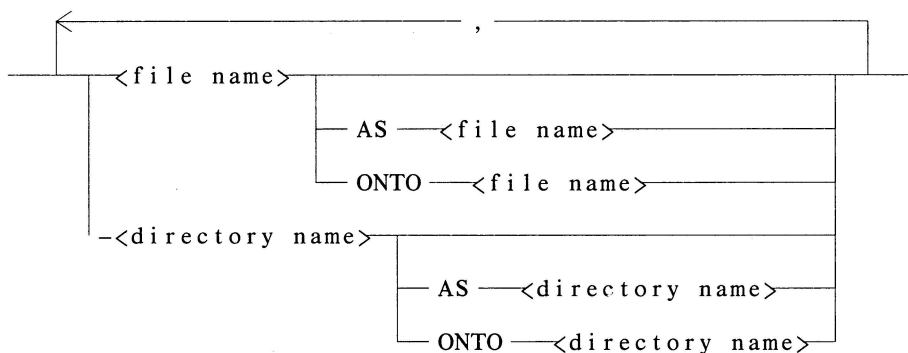
```
c a/b as $test/a/b:dcalgol
#COMPILING 3725
#ET=29.8 PT=5.6 IO=2.3
```

```
COMPILE A ON B AS Z WITH MYCOMP ON MYPACK
#COMPILING 3727
#ET=1:28.6 PT=46.3 IO=36.4
```


Example

```
CONT ?
#NO CONTINUATION CHARACTER DEFINED
CONT = *
#CONTINUATION CHAR =*
TERM WIDTH 28
#LINE = 28, PAGE = 24, BUF =1920, SCREEN, WAIT, " "
MAKE*
#%
A*
#%
ALGOL
#WORKFILE A: ALGOL
100 ENTER A LONG LINE ON A*
#%
NARROW SCREEN
LIST
100 ENTER A LONG LINE ON
    A NARROW SCREEN
#
CONT
#
CONT ?
#NO CONTINUATION CHARACTER DEFINED
```

CANDE REFERENCE MANUAL

COPY**Syntax****<copy file list>****Semantics**

The COPY command is recognized by CANDE as a Work Flow Language (WFL) command. CANDE passes the COPY specifications to the WFL compiler for processing in the same manner as for the WFL command.

The COPY command is used to copy files to and from disk, tape, and disk pack. If a file at the destination has the same title as any file being copied, the file at the destination is implicitly removed, and the new file is copied to the destination. For more detailed semantical information and a further breakdown of syntax, refer to the COPY statement in Section 6 of the B 5000/B 6000/B 7000 Series WFL Reference Manual, form 5011794.

This command may not be used for files in another user's library.

Examples

COPY A TO B(PACK)

#RUNNING 3794

#BOT 3795 (UZER)WFLCODE

#BOT 3796 LIBRARY/MAINTENANCE

#3796 (UZER)A COPIED.

#EOT 3796 LIBRARY/MAINTENANCE

#EOT 3795 (UZER)WFLCODE

#

COPY & COMPARE A AS B FROM DISK TO MASTERPACK(PACK)

#RUNNING 3842

#BOT 3843 (UZER)WFLCODE

#BOT 3844 LIBRARY/MAINTENANCE

#3844 (UZER)CANDE COPIED.

#EOT 3844 LIBRARY/MAINTENANCE

#EOT 3843 (UZER)WFLCODE

#

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES. |

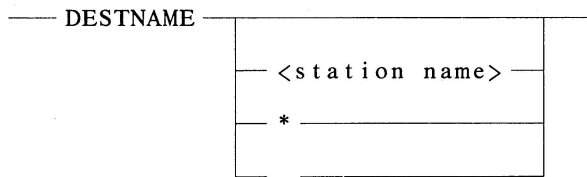
CANDE REFERENCE MANUAL

DC CLUSTER 4,2
#RUNNING 3966
CLUSTER 4,2
SYSTEM/DCSTATUS (III.0.90) DATE : 04/20/78 TIME : 1555:50
FILES: ELMONTENDL/NIF ELMONTENDL/DCPCODE
DATE OF NDL COMPILATION : 03/03/78 , TIME : 0724:44
NO RECONFIGURATION EXECUTED.
*--DCP IS NOT NDL DEFINED --- 4 ---
NORMAL TERMINATION

#

DCSTATUS GRAPH;FILE LINE(KIND=PRINTER)
#RUNNING 40000
#

CANDE REFERENCE MANUAL

DESTNAME**Syntax**

The DESTNAME command provides a means of specifying a non-CANDE station for the printed output of a task. This command specifies a DESTNAME for a complete session, automatically supplies that DESTNAME when a task is executed, and controls the setting of the automatic DESTNAME. The minimum abbreviation for the DESTNAME command is DEST.

The DESTNAME default is supplied on all task executions explicitly initiated by the user and on CANDE-initiated tasks, such as, CANDE WRITE. The DESTNAME may be changed for some portions of a session, may return to the default value, or may be interrogated for the current value. One possible value is null; that is, the DESTNAME value may be set so that no DESTNAME is supplied for a task execution. The current DESTNAME may also be overridden by an explicit specification on a RUN or COMPILE request.

When the DESTNAME command is the only input entered, CANDE reports the current value of the DESTNAME.

If the <station name> is supplied, it is saved by CANDE as the current value until another DESTNAME action takes place.

The input "*" causes the current value of the DESTNAME to return to the default of the user associated with the session.

The input "." causes the DESTNAME to be set to null. In this case, DESTNAME is not supplied by CANDE on any task execution request.

If the DESTNAME is explicitly changed during a session, back-up requests (such as BDREMOVE and BACKUPPROCESS) search for BD, BP, REMLP<nn>, and REMCP<nn> files, where <nn> is the NDL-defined MCS number for each destination station used during that session. Searches are made on DISK, PACK, and the DL BACKUP device that was in effect at the beginning of the session. The message NO BACKUP FILES FOUND is emitted if no such files are found.

Because the DESTNAME for a session is associated with a user, the default DESTNAME is kept in the USERDATAFILE. If the permanent value of the DESTNAME is to be changed in the USERDATAFILE, SYSTEM/MAKEUSER may be invoked by the installation. (Refer to the B 5000/B 6000/B 7000 Series System Software Operational Guide, Volume 2, Chapter 9 for further information.) The USERDATAFILE entry is made using the word CANDEDESTNAME, which is of type FILE (that is, a station name has the form of a file title).

Changing DESTNAME in RUN/EXECUTE Statements

When the value of DESTNAME is explicitly changed in a RUN or EXECUTE statement, the new value overrides the former value for the session for that request only. No abbreviation for DESTNAME is accepted in a RUN or EXECUTE command. (See the RUN and EXECUTE statements in this section for the correct syntax.) Changing the DESTNAME for execution of any given task in the session does not affect the default DESTNAME for the session.

Examples

DESTNAME.

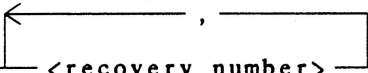
DEST RJE

DEST*

DESTNAME

CANDE REFERENCE MANUAL

DISCARD**Syntax**

— DISCARD —  —————

Semantics

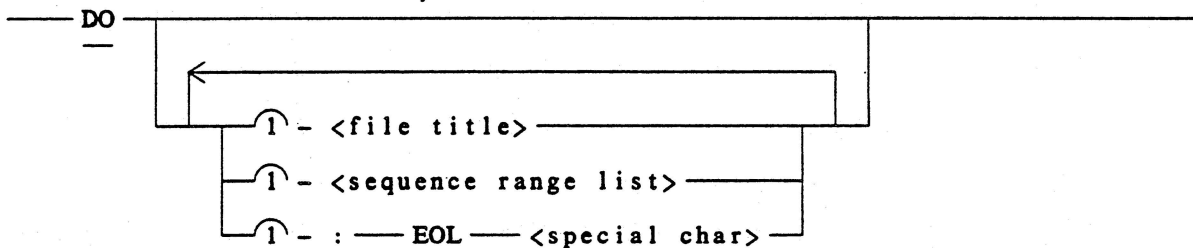
The DISCARD command allows removal of unwanted recovery files. The keyword must be followed by a recovery file number or a list of recovery file numbers.

Examples

```

REC
#RECOVERY DATA ON USERPACK
 150 A (04/24/78)
 151 Z (04/24/78)
#
DISCARD 150
#
REC
#RECOVERY DATA ON USERPACK
 151 Z (04/24/78)
#
DIS 151
#
REC
#

```

DO**Syntax****Semantics**

Use of the DO command causes a file (or portions thereof) of CANDE input to be presented. This input is performed synchronously in the session. Valid input includes CANDE commands (except control commands) and single-line insertions.

The specified input is entered in the user's queue and performed on completion of the DO command. This queued input is treated as any other queued input except that each command is displayed when it is removed from the user's queue. When this queue is displayed with the ?SHOW control command, an entry from a DO command is preceded by a "D".

If no <file title> is specified, the user's workfile is used. If no <sequence range list> is specified, the entire file is used. Only the text field of the file is significant, and trailing blanks are removed. If trailing blanks are significant, an end-of-line character may be specified (via the :EOL syntax) to allow specification of significant characters. The syntax and semantics of the :EOL feature are the same as for the SCHEDULE command.

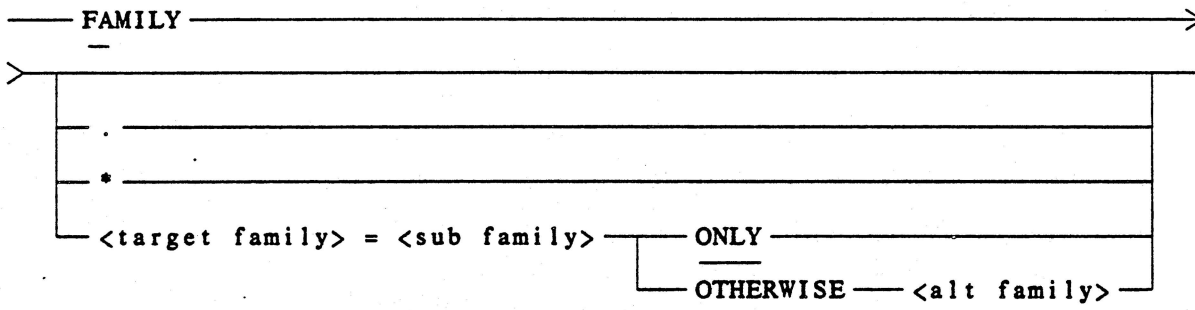
Examples

```
LIST X
#FILE (UZER)X ON USERPACK
100 WHAT;REM;REM Y
200 MAKE Y ALGOL
300 100TEXT FOR Y
400 200LIST Y
500 300MORE TEXT FOR Y
600
700 SAVE;L
800 DO 200
900 WHAT
#
DO X
#FILE (UZER)X ON USERPACK
#
WHAT;REM;REM Y
#WORKFILE TEMP: SEQ, 5 RECORDS (THRU 5000), SAVED
#
# (UZER)Y ON USERPACK REMOVED
MAKE Y ALGOL
#WORKFILE Y: ALGOL
100TEXT FOR Y
200LIST Y
300MORE TEXT FOR Y

#
SAVE;L
#UPDATING
#WORKSOURCE Y SAVED
100 TEXT FOR Y
200 LIST Y
300 MORE TEXT FOR Y
#
DO 200
#WORKFILE Y
#
LIST Y
#FILE (UZER)Y ON USERPACK
100 TEXT FOR Y
200 LIST Y
300 MORE TEXT FOR Y
#
WHAT
#WORKFILE Y: ALGOL, 3 RECORDS (THRU 300), SAVED
```


FAMILY

Syntax



Semantics

The FAMILY command allows alteration of the family specifications that are to be applied to the current session. Default family specifications are obtained by CANDE from the USERDATAFILE.

Use of the FAMILY command causes CANDE to display the FAMILY currently in effect for the session.

Use of the FAMILY. command deletes the family specifications.

Use of the FAMILY* command causes CANDE to restore the current family specifications to the USERDATAFILE default.

If the FAMILY command changes the <target family> from the current <target family> and an unsaved workfile exists, CANDE displays the following query:

```
#REMOVE OR SAVE WORKFILE
```

The unsaved workfile must either be removed or saved, and the command must be re-entered.

<target family>, <sub family>, and <alt family> are all <family name>s.

The FAMILY specification is used to redirect references to files on the <target family>. Requests for an existing file on the <target family> search <sub family> and then <alt family> (if it exists). New files that request the <target family> are located on <sub family>.

When creating or referencing files through CANDE, the files are always associated with the family DISK. Family substitution is effective for CANDE functions only if <target family> is DISK. Thus, a family specification of PACK=X ONLY does not affect files referenced by CANDE, but affects tasks that reference files on the family PACK. If the <target family> is not DISK, then CANDE creates all files on the family DISK.

Examples

FAM DISK=PACK ONLY
#FAMILY DISK = PACK ONLY

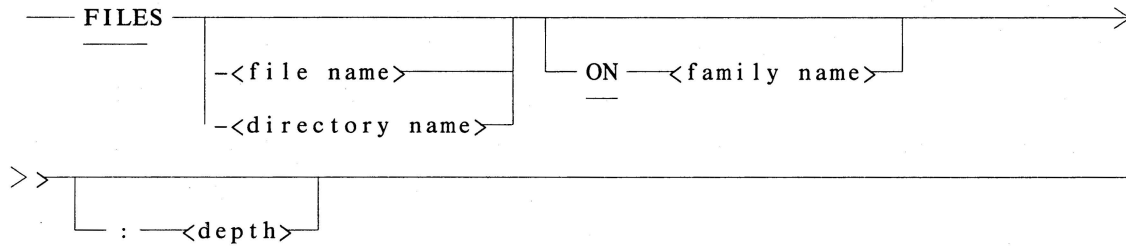
FAMILY.
#FAMILY.

FAM*
#FAMILY DISK = USERPACK OTHERWISE DISK

FAMILY X=Y OTHERWISE Z
#FAMILY DISK = Y OTHERWISE DISK

FILES

Syntax



Semantics

The FILES command causes CANDE to list the names and types of files in the user's library. If no qualifier is given, all files and directories in the user's library on the <family name> of the current workfile are listed. If a <directory name> is specified, the names and types of files under that <directory name> for the current <family name> are listed. If a <file name> is given, the name and type of the file on the current <family name> are listed. If the name supplied is both a <file name> and a <directory name>, all files under the <directory name> and the <file name> on the current <family name> are listed.

The <depth> option may be supplied to indicate the level of retrieval of names to be listed.

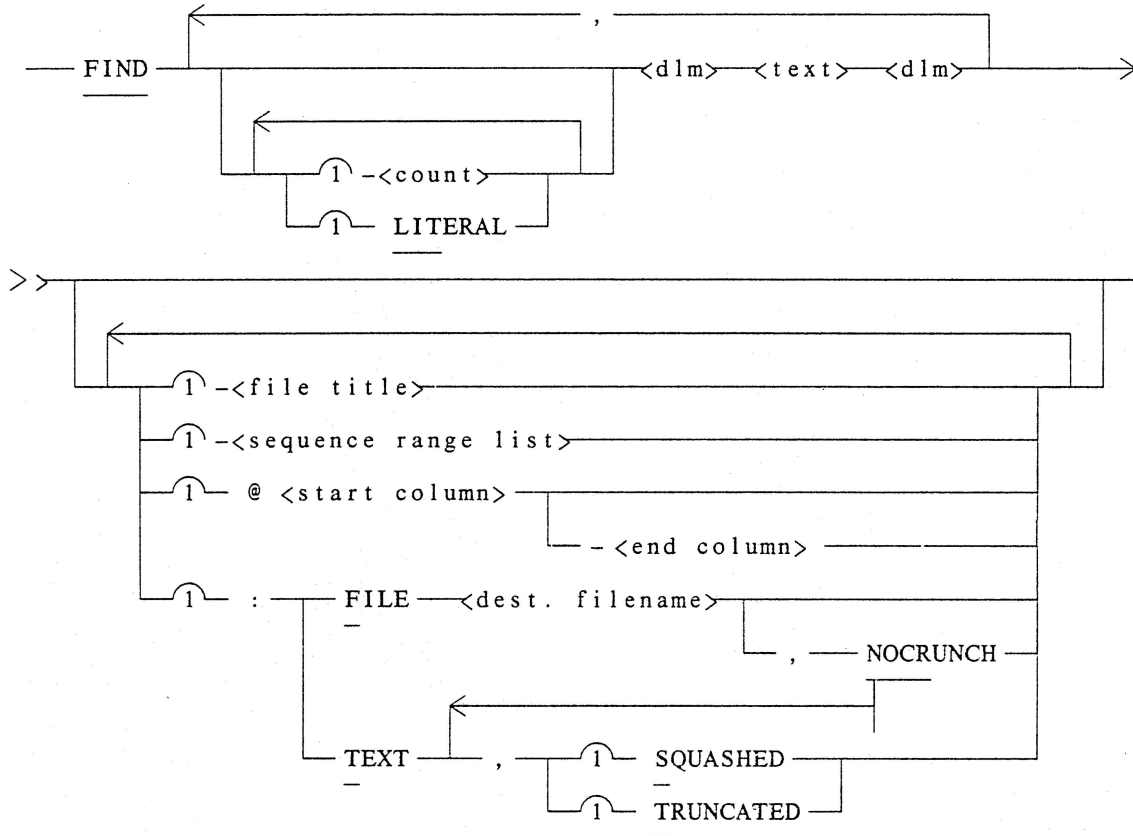
Examples

```

FILES
(AVERAGEUSER) ON DISK
.  AFILE
.  .  A1 : SEQDATA
.  .  A2 : SEQDATA
.  BFILE
.  .  B1 : SEQDATA
.  .  B2
.  .  THREEDEEP : SEQDATA
.  CFILE : SEQDATA
#
  
```

```

files : 2
(AVERAGEUSER) ON DISK
.  AFILE
.  .  A1 : SEQDATA
.  .  A2 : SEQDATA
.  BFILE
.  .  B1 : SEQDATA
.  .  B2
.  CFILE : SEQDATA
#
  
```

FIND**Syntax****Semantics**

The FIND command searches a file (by default, the workfile) for appearances of specific target text. The line numbers of lines where the target text is found are displayed at the user's terminal with an asterisk (*) denoting line numbers where the target text appears more than once. The output option TEXT displays the lines of text in addition to the line numbers. The FILE output option directs the lines of text to a new file instead of the terminal.

<text> specifies the target text and must be bounded by two matching delimiters, <dlm>. In a single FIND command, one or more target texts can be the subject of the search. <text> can contain any characters except the <dlm> and cannot be null.

<count> must be an unsigned <integer>; it limits searching to the first <count> occurrences of the target. All occurrences after that number are ignored. If more than one target text is specified, <count> applies only to its associated <text>.

CANDE REFERENCE MANUAL

LITERAL specifies that the mode of the search for the associated <text> is "literal mode". The default, when LITERAL is not specified, is "token mode". (Refer to the pragmatics of this command for an explanation of search modes.)

<file title> specifies the file to be searched and can be any file the user is allowed to access. When <file title> is not specified, the workfile is searched.

<sequence range list> restricts the lines to be searched to one or more sequence ranges.

The (@) option restricts the search to a column range on each line. <start column> specifies the column number where the search is to begin. <end column> specifies the column number where the search is to be terminated. The column range can specify any part of the usable line except the sequence number field (for example, "@81-90" is valid for type ALGOL files). The entire text field of the line is searched if no columns are specified. If only <start column> is specified, the default <end column> is the last column of the text field. Thus, to search only column 1 of each line, the specification would be "@1-1".

The TEXT option outputs the entire line containing the target text. SQUASHED causes a sequence of blanks to be printed as a single blank. TRUNCATED causes the output line to be truncated to the character width of the terminal if necessary. By default, lines that exceed terminal width are split across two or more lines.

The FILE option outputs the entire line containing the target text to a new file in the user's library. <dest. filename> specifies the name of that new file. The new file is the same type as the workfile or <file title>. Files created by the FIND command are crunched files. If a crunched file is not desired, the output option NOCRUNCH may be specified.

<dlm> may be any special character except semicolon (;), comma (,), at sign (@), or colon (:).

Pragmatics

Two text searching modes are employed by CANDE. Both the FIND and REPLACE commands use these searching mechanisms. A token mode search is performed by default, and the literal mode search can be invoked by using the LITERAL option.

In literal mode, each line of the file is considered an arbitrary string of characters, including blanks. The search is successful whenever the sequence of characters in the target text exactly matches a sequence of characters in a line of the file. For example, if the target is lit .a - b. (periods are used as <dlm>s), the following occurrences are examples of matching target text:

```
x=a - b
x=aa - b
x=a - bb
```

However, the same target lit .a-b. (periods are used as <dlm>s) does not match the following occurrences of target text:

```
x= a - b
x= aa - b
x= a - bab
```

CANDE REFERENCE MANUAL

In token mode, each line is considered as a sequence of tokens. A token is defined as either:

1. Any group of adjacent alphanumeric characters (letters a through z, and digits 0 through 9), or
2. Any nonalphanumeric graphic character except blank.

Any number of blanks can separate tokens; at least one blank must separate adjacent alphanumeric tokens. For example, the following line illustrates the use of tokens:

```
100 abc, def5-3xyz i j k& *+@
```

The line above contains the following tokens:

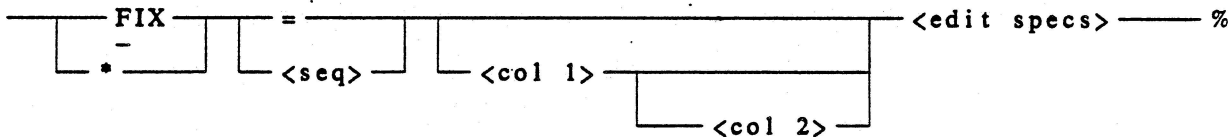
```
abc
,
def5
-
3xyz
i
j
k
&
*
+
@
```

The search is successful whenever the same sequence of tokens in the target text is found in a line of the file. As examples, any of the following targets would be found in line number 100 referenced above (periods are used as <dlm>s):

```
.abc.
. , def5      -      3xyz.
.*           +      @.
.&*.
.-      3xyz.
```

Examples

```
LIST FILE1
#FILE (UZER)FILE1 ON USERFILES
100 THIS IS A LIST.
200 THIS IS A PAGE.
300 HERE IS SOME CODE.
400 SECOND LIST.
500 END.
#
FIND /LIST/:T
#WORKFILE FILE1
100 THIS IS A LIST.
400 SECOND LIST.
#
FIND /LIST/,/PAGE/,/CODE/ 300-END
#WORKFILE FILE1
300, 400
#
```

FIX**Syntax****Semantics**

The basic action of a FIX command is to locate an "objective", which may be a character or group of adjacent characters to be deleted or replaced; the objective may also be a point between two characters where an insertion is to be made. When the objective is found, the deletion and/or insertion is performed. Text to the right of the objective is moved right or left if the insertion is longer or shorter than the deletion; trailing blanks are discarded or inserted as needed. If nonblank characters are pushed past the end of the text field, the FIX operation is performed, but an overflow error is noted. Any information in an ID field (such as a patch mark) is set to blanks when a line is FIXed. The asterisk (*) is a synonym for FIX and may be used to edit lines while in automatic sequence mode. If the asterisk is the first character after the system-supplied sequence number, CANDE scans the line to determine if it is a valid FIX command. If the line is valid, the command is executed; otherwise, the line is entered as a line of text.

The FIX command is one of the commands that are "tanked" and are executed only when the next UPDATE occurs. Because of tanking, any error messages that result are not displayed until the time of the next UPDATE or LIST; therefore, the error messages are not displayed at the time the command is entered.

The sequence number of the line to be fixed is specified as an integer <seq> or by an equal sign (=); an equal sign refers to the last line entered or fixed and is undefined if no entry or FIX has been entered since the workfile was updated.

The target search may be bounded or supplanted by column specifications; <col 1> and <col 2> represent integers; <col 2> may not be less than <col 1>, and both must lie within the text field for the workfile.

CANDE REFERENCE MANUAL

The keywords BEFORE or AFTER may appear in the single-target FIX. INCLUSIVE or EXCLUSIVE must appear in the dual-target fix. Each <dlm> must be the same delimiter [which may be any nonblank, nonalphanumeric character except dash (-) or semicolon (;)]. A target (<target>, <target1>, or <target2>) may contain any sequence of characters except the <dlm>; the target is null if its bounding delimiters are in adjacent columns of the input record.

The <new text> may contain any sequence of characters and is null if the end-of-line (<eol>) immediately follows the last <dlm>. (In a SCHEDULE session, the end-of-line is assumed to follow the last nonblank character on the line, unless an EOL character has been defined and used.)

If the target in a single-target FIX is non-null, CANDE scans the text field of the specified line in the workfile until a sequence of characters is found that exactly matches the target. All characters including blanks are significant. If no keyword appears, this target is the objective. If BEFORE or AFTER appears, the objective is the point just preceding or just following the target, respectively. If both targets in a dual-target fix are non-null, the first target is sought as above, and then the second is sought in the remainder of the text field beyond the first target. The objective is then the aggregate of these two targets and the text between them (INCLUSIVE) or just the text between them (EXCLUSIVE). If <col 1> appears, searching begins at <col 1> rather than the beginning of the text field. If <col 2> also appears, no target is sought which extends past <col 2>. If a target is null, no search is made; <target> or <target1> is "found" at the beginning of the text field (or at <col 1> if specified); <target 2> is "found" at the end of the text field (or <col 2> if specified). Column specifications affect determination of the objective but do not limit the text that may be shifted to effect the fix.

The following listing summarizes the many variations of the FIX command. In the following listing, m and n represent integer column numbers; x and y represent text strings. A blank space in the listing denotes an absent <col 1>, <col 2>, or keyword; a hyphen denotes a null target. Where several variations achieve the same effect, only the simplest variation is shown (degenerate cases like BEFORE or AFTER an empty target are omitted). If <new text> is null, the word "delete" should be "replace", and the words "do nothing" should be "insert."

CANDE REFERENCE MANUAL

```

<col 1>
| <col 2>
| | <keyword>
| | | <target> or <target 1>
| | | | <target 2>
| | | | |
| | | | | Summary of Effect
| | | | |
| | | | | - Insert at beginning.
m | | | | - Insert at column <m>.
m n | | | | - Replace columns <m> through <n>.
| | | | x Replace <target>.
m | | | | x Replace <target> found in <m> through end-of-text.
m n | | | | x Replace <target> found in <m> through <n>.
| | | | B x Insert before <target>.
| | | | A x Insert after <target>.
m | | | | B x Insert before <target> found in <m> through end-of-text.
m | | | | A x Insert after <target> found in <m> through end-of-text.
m n | | | | B x Insert before <target> found in <m> through <n>.
m n | | | | A x Insert after <target> found in <m> through <n>.

| | | | I - - Replace entire text field.
m | | | | I - - Replace text from column <m> onward.
| | | | I - x Replace text through <target2>.
| | | | E - x Replace text prior to <target2>.
m | | | | I - x Replace from column <m> through <target2>.
m | | | | E - x Replace from column <m> until <target2>.
m n | | | | I - x Replace from column <m> through bounded <target2>.
m n | | | | E - x Replace from column <m> until bounded <target2>.
| | | | I x - Replace <target1> and rest of text.
| | | | E x - Replace text beyond <target1>.
m | | | | I x - Replace bounded <target1> and beyond.
m | | | | E x - Replace beyond bounded <target1>.
m n | | | | I x - Replace bounded <target1> through column <n>.
m n | | | | E x - Replace after bounded <target1> until column <n>.
| | | | I x y Replace <target1> through <target2>.
| | | | E x y Replace between <targets>.
m | | | | I x y Replace as above with
m | | | | E x y <targets> bounded in <m> to end of text.
m n | | | | I x y Replace as above with
m n | | | | E x y <targets> bounded in <m> to <n>.

```

NOTES

1. The target search in a FIX command is satisfied by the first match encountered. Enough context (or a column restriction) must be provided to unambiguously specify the intended objective.
2. The <dlm> character may not appear in a target, but it may appear in the new text. Inadvertent use of <dlm> within <target> is not detected as an error; CANDE processes a shorter <target> and a longer <new text> than is intended.

CANDE REFERENCE MANUAL

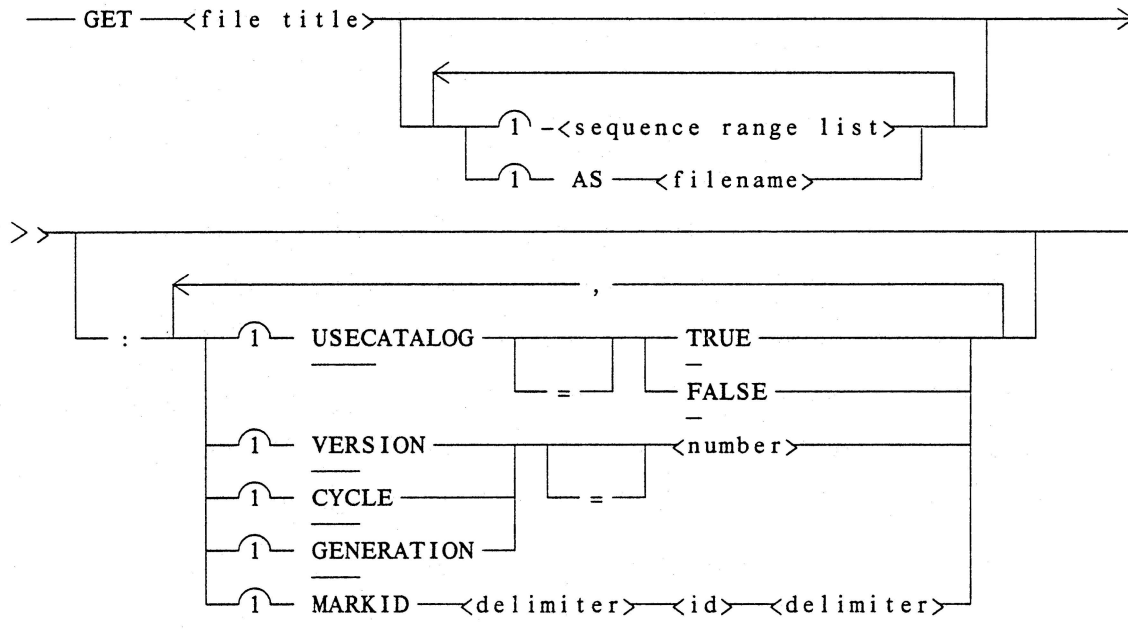
Examples

```

L
100 ABCDEFGHIJKLMNOPQRSTUVWXYZ
300 IF CARD.PRESENT THEN
400 VIVID HUES OF RED, ORAGE, GREEN AND BLUE
500 X = SQRT (3.1459 * R**2)
#
FIX 100:FG:HIJ
#
L 100
100 ABCDEHIJHIJKLMNOPQRSTUVWXYZ
#
FIX 100 E.HIJ..
#
L 100
100 ABCDEHIJ
#
F 300/CARD.PRESENT/BOOLEAN(CARD.AVAILABLE)
#
L 300
300 IF BOOLEAN(CARD.AVAILABLE) THEN
#
F 300.CARD.PRESENT.BOOLEAN(CARD.AVAILABLE)
#
L 300
300 IF PRESENT.BOOLEAN(CARD.AVAILABLE).PRESENT THEN
#
F 400 A.NGE., YELLOW
#
L 400
400 VIVID HUES OF RED, ORAGE, YELLOW, GREEN AND BLUE
#
FIX 400 20-50//
#
L 400
400 VIVID HUES OF RED,
#
FIX 500 E/(/)/AREA
#
L 500
500 X = SQRT (AREA)
#

M TEST
#WORKFILE TEST:SEQ
S
100THIS IS A TEST
200* 100/IS/ISNT
300
#
#
L
100 THISNT IS A TEST
#

```

GET**Syntax****Semantics**

The GET command designates an existing saved file, or portions thereof, as the source for a new workfile. If an unsaved workfile exists, the command is ignored, and the following error message is displayed:

```
#REMOVE OR SAVE WORKFILE
```

<file title> can specify any file to which the user is permitted access. If desired, only a portion or portions of a file are used as the source if a <sequence range list> is specified.

The name of the workfile to be created can be specified by using the AS <file name> option; otherwise, the workfile name is assumed to be a temporary copy of <file title>.

For installations that use file cataloging, the GET command allows specification of USECATALOG, VERSION, CYCLE, and GENERATION. For an explanation of these attributes refer to the B 7000/B 6000 Series I/O Subsystem Reference Manual, form 5001779.

The MARKID <delimiter> <id> <delimiter> option establishes an initial markid. The MARKID command may still be used at any time afterwards to establish a new markid.

CANDE REFERENCE MANUAL

The following commands display the correspondence between the ID field of a record and the current value of the markid, if the MARKID option is specified:

1. FIND on a workfile using the TEXT option.
2. RANGE on a workfile using the TEXT option.
3. LIST (or PRINT) on a workfile without the UNSEQUENCED or PUNCH options.
4. REPLACE using the TEXT option, before the record ID field is replaced.

The displays from the preceding commands include the following:

1. The sequence number.
2. An asterisk ("*") if the record id is the same as the mark id; otherwise, a blank (" ").
3. The text.

Examples

```
GET AFILE
#WORKFILE AFILE: SEQ, 5 RECORDS, SAVED
```

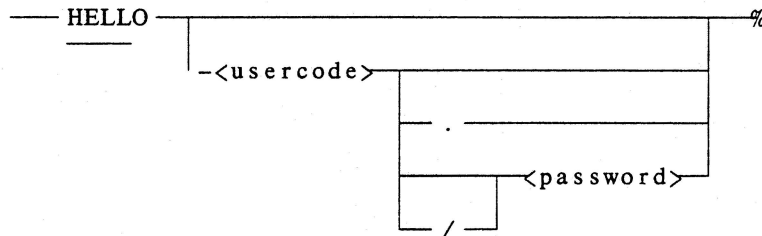
```
LOAD TERMPAPER 400-END AS INDEX
#WORKFILE INDEX; SOURCE IS (UZER)TERMPAPER ON USERPACK: DATA
```

```
GET (ANYUSER)ANYFILE ON HISPACK AS MYWORKFILE 0-150,400-END
#WORKFILE MYWORKFILE; SOURCE IS (ANYUSER)ANYFILE ON HISPACK: ALGOL
```

```
GET EXAMPLE: MARKID-10/01/81-81-
#WORKFILE EXAMPLE: ALGOL,500 RECORDS, SAVED, MARKID "10/01/81 "
```

HELLO

Syntax



Semantics

The HELLO command terminates the current session and initiates a new session under a different usercode. Dial-up lines are not disconnected.

Usercode and password can be included to complete the log-in procedure, or this information can be requested in the normal log-in manner. If an unsaved workfile exists at the time that the HELLO is entered, CANDE displays the following query:

```
#REMOVE OR SAVE WORKFILE
```

This query indicates the presence of the unsaved workfile. The workfile must be removed or saved and the HELLO command re-entered before the session can be terminated.

The <usercode> . form is used for a usercode without a password.

System resource usage statistics are provided on termination of the session. The session number, elapsed time of session (ET), processor time consumed (PT), I/O time consumed (IO), usercode, current time, and the date are printed in the following format:

```
#END SESSION <session number> ET=<time> PT=<time> IO=<time>
#USER = <usercode> <time> <date>
```

If a chargecode is used for the session, the charge code is printed following the usercode identification.

Examples

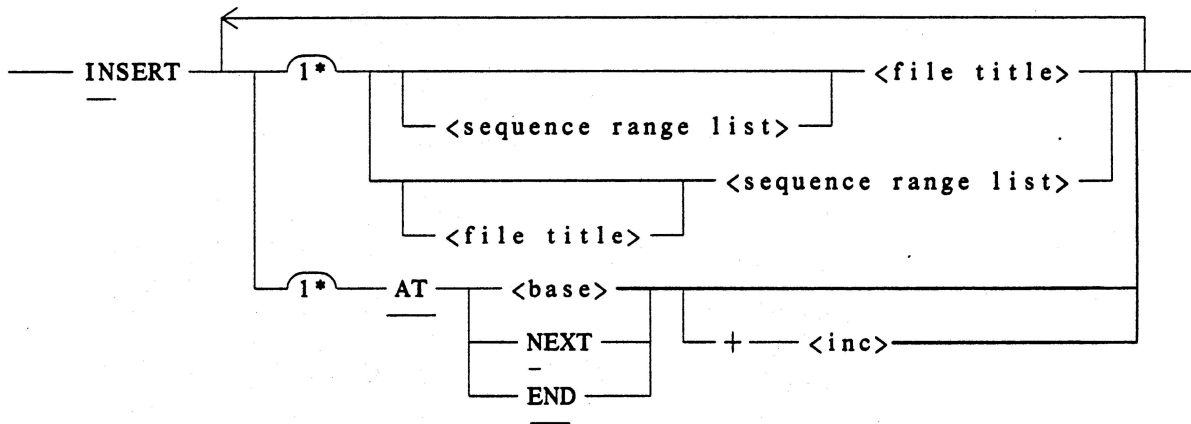
```
HELLO
#B6800:277 CANDE 30.90 YOU ARE DOCUMENT1(13)
#ENTER USERCODE PLEASE
```

```
hell userid
#ENTER PASSWORD PLEASE
```

```
HELLO USERID PASSID
#SESSION 3975 16:30:56 04/20/78
```

INSERT

Syntax



Semantics

The INSERT command copies lines from a file (by default, the workfile), and places the copies in the workfile as a contiguous block with new sequence numbers.

<file title> names the file to be inserted and can be any file to which the user is allowed access. The workfile is assumed if <file title> is unspecified.

<sequence range list> specifies the sequence range(s) of the lines to be copied into the workfile.

The new sequence numbers of copied lines are determined by assigning an initial value to the first line and incrementing that value for each succeeding line. The initial value must be specified in one of three ways:

1. <base> specifies the value explicitly. <base> must be an <integer>.
2. NEXT sets the initial value to the next sequence number that would have resulted from the most recent MOVE, INSERT, RESEQ, or SEQ command. A default value of 100 is used if no such command has appeared since the last MAKE, GET, or DELETE ALL command.
3. END sets the initial value to the largest sequence number currently in the file, plus the current specified (or default) increment.

CANDE REFERENCE MANUAL

<inc> defines the increment value to use for successive sequence numbers in the block. <inc> must be an <integer>. If <inc> is unspecified, the increment value from the most recent MOVE, INSERT, RESEQ, or SEQ command is used. (100 is used if none of these commands have appeared.)

The range of new sequence numbers cannot overlap any lines already in the file, nor can the numbers exceed the largest sequence number that may be expressed in the sequence number field.

Examples

```
LIST
#WORKFILE TESTONLY
100 LINE 1
200 LINE 2
300 LINE 3
400 LINE 4
#
```

```
INSERT 100-300 AT END+10
#UPDATING
#
```

```
LIST
#WORKFILE TESTONLY
100 LINE 1
200 LINE 2
300 LINE 3
400 LINE 4
410 LINE 1
420 LINE 2
430 LINE 3
#
```

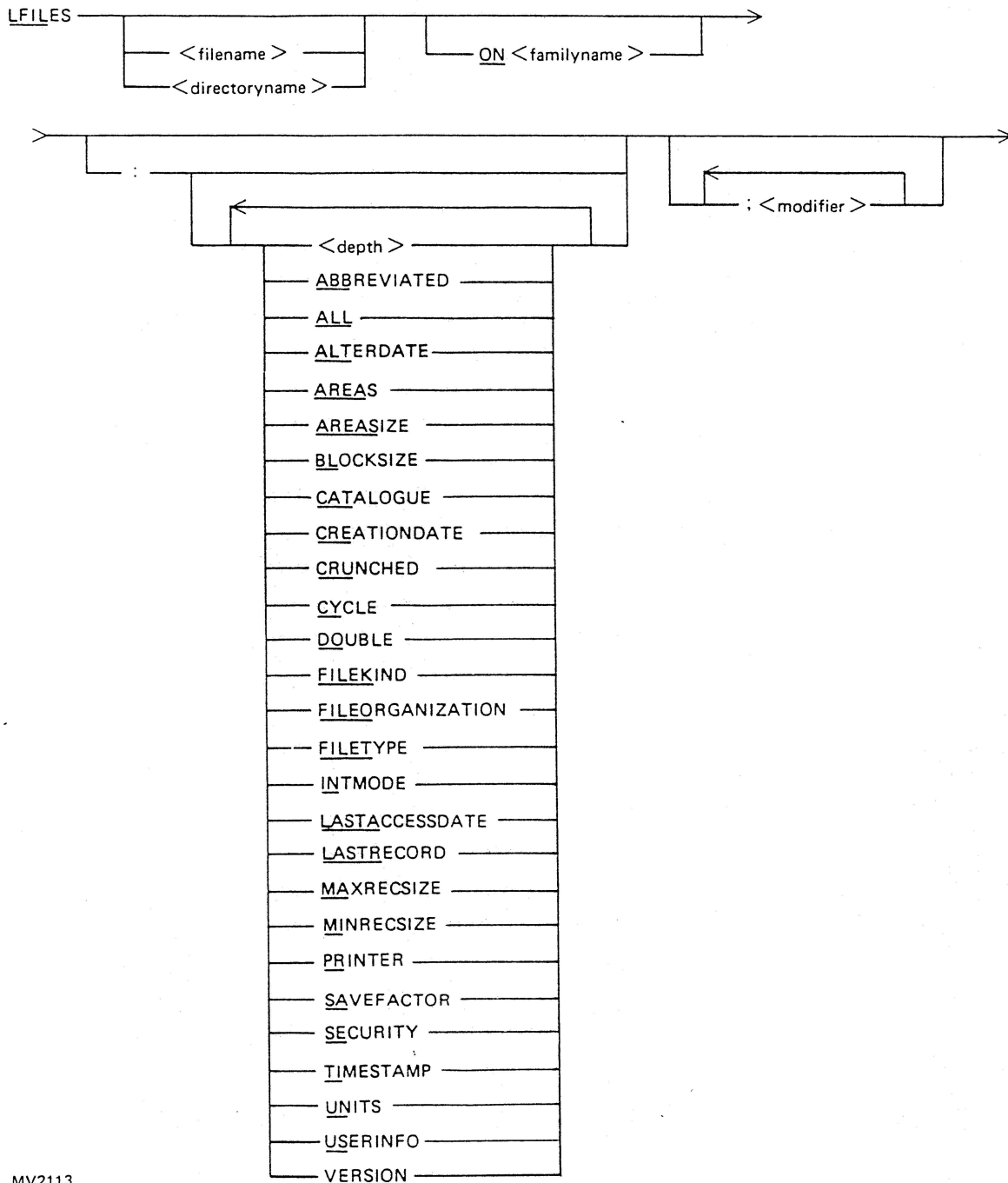
```
INS VEGETABLES AT 402+2
#UPDATING
#
```

```
INSERT AT END FRUIT
#UPDATING
#
```

```
LIST
100 LINE 1
200 LINE 2
300 LINE 3
400 LINE 4
402 CARROT
404 TOMATO
406 EGGPLANT
410 LINE 1
420 LINE 2
430 LINE 3
432 BANANAS
434 ORANGES
436 APPLES
```

LFILES

Syntax



MV2113

Semantics

The LFILES command results in the execution of the system utility SYSTEM/FILEDATA, which lists the names and file attributes of files. If a <file name> or <directory name> is not specified, the names and attributes of all files in the user's library are listed.

<file name> names a specific file for which file attribute information is desired. <file name> can refer to any file to which the user is allowed access.

<directory name> names a specific directory and results in a list of the file attribute information for each file subordinate to that directory. However, information is printed only for those files to which the user is allowed access.

If both a file and a directory exist for the name supplied, information is listed for both the file and the directory (plus the subordinate files of the directory).

The amount of file attribute information displayed depends on whether available options are ON or OFF. By default, all information regarding file attributes is displayed (that is, all options are ON). The amount of file attribute information to be output can be restricted by naming attribute options and using the available output options. Options are recognized following the appearance of the colon. The colon turns off all file attribute options except FILEKIND; the file attribute options can then be turned on selectively.

The file attribute options are as follows:

- ALTERDATE
- AREAS
- AREASIZE
- BLOCKSIZE
- CREATIONDATE
- CRUNCHED
- CYCLE
- FILEKIND
- FILEORGANIZATION
- FILETYPE
- INTMODE
- LASTACCESSDATE
- LASTRECORD
- MAXRECSIZE
- MINRECSIZE
- SAVEFACTOR
- SECURITY
- TIMESTAMP
- UNITS
- USERINFO
- VERSION

Some of the file attribute option names are identical to the corresponding file attribute names but print more information than the file attributes contain.

The file attribute options ALTERDATE, CREATIONDATE, and LASTACCESSDATE print both the date and the time.

CANDE REFERENCE MANUAL

The output options available are discussed in the following paragraphs.

<depth>

<depth> must be an unsigned <integer>. This option restricts the depth of listing for multilevel file names.

ABBREVIATED

The ABBREVIATED option causes the specified file attribute names to be abbreviated in the output.

ALL

The ALL option causes information for all file attributes to be listed. This option is assumed if no options are specified.

CATALOGUE

The CATALOGUE option reports the existence of nonresident, catalogued files.

DOUBLE

The DOUBLE option causes the generated output to be double-spaced.

PRINTER

The PRINTER option causes the output to be sent to the line printer.

Examples

```
LFILES BREAKER ON USERFILES
```

```
#RUNNING 2497
```

```
#?
```

```
(TERRI) : DIRECTORY ON USERFILES
```

```
. BREAKER : ALGOLSYMBOL AREAS=15 AREASIZE=450 BLOCKSIZE=450
      CREATIONDATE= 2/12/81 @ 13:19:15 CYCLE=1 FILETYPE=0
      INTMODE=4 LASTACCESSDATE= 2/12/81 @ 13:19:15
      LASTRECORD=12 (225 SEGS) MAXRECSIZE=15 MINRECSIZE=0
      SAVEFACTOR=30 SECURITY=PRIVATE (I/O)
      TIMESTAMP= 2/12/81 @ 13:19:17 UNITS=0 VERSION=0
      ALTERDATE= 2/12/81 @ 13:19:15
```

```
#
```

```
LFILES (JOHNSON)CSG/IOSTANDARDS:BLOCKSIZE AREASIZE MAXRECSIZE ABB
```

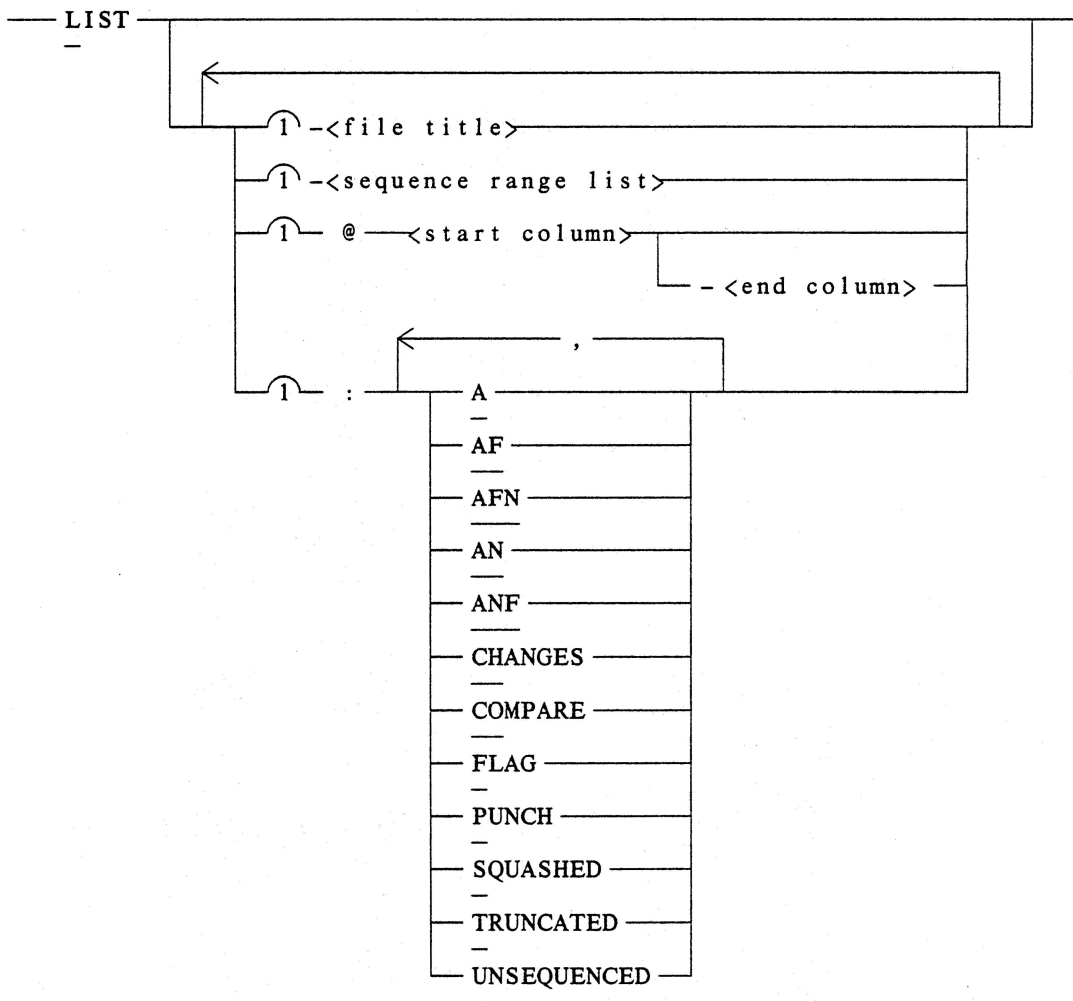
```
#RUNNING 2513
```

```
#?
```

```
(JOHNSON) : DIRECTORY ON USERPACK
```

```
. CSG : DIRECTORY
. . IOSTANDARDS : SEQDATA ASIZE=420 BLK=420 MAX=14
```

```
#
```

LIST**Syntax****Semantics**

The LIST command displays the contents of the workfile or some other file at the terminal. If <file title> is not specified, the contents of the workfile are listed. If a <file title> is named, the file is listed if the user has access privileges.

A <sequence range list> in the command indicates that only the records in the <sequence range>s specified are to be listed.

A <column range>, which is (<start column> (<end column>)), indicates that only the specified columns of each record are to be listed.

CANDE REFERENCE MANUAL

Format Options

Several output format options are available with the LIST command. If no format option is specified, the default format is used; that is, the sequence number of the line appears (pseudo sequence numbers for type data files) with leading zeros omitted, followed by a blank and the text of the line. Lines too long for the terminal are split (on token boundaries where possible). The format options and their effects are described in the following paragraphs.

PUNCH

The function of the PUNCH option is to punch a paper tape on the paper tape punch of a terminal so equipped. The system waits five seconds for the user to turn on the paper tape punch if the punch is not in auto-start. The system sends a DC1 (XON) character followed by 10 NULs, and the name of the file (or the workfile name if <file title> is not specified), followed by 40 additional NULs. Each line of data is ended with a carriage return, a line feed, DC1 character, and a NUL. A DC3 (XOFF) character is punched at the end of the tape. The tape can be read back to the system using the TAPE command by positioning the tape in the reader in the area of the 40 NULs. Sequence numbers punched include leading zeros.

SQUASHED

The SQUASHED option causes any group of multiple blanks to be reduced to a single blank.

TRUNCATED

The TRUNCATED option causes the output line to be truncated if the width of the CANDE station is less than the MAXRECSIZE of the workfile. (The continuation character designates the character position at which truncation takes place.)

UNSEQUENCED

The UNSEQUENCED option causes the lines to be listed without sequence numbers.

Altered Records List Options

The following LIST options are used to list alterations that have been made to the workfile by FIX, DELETE, and single-line entry since the workfile was last updated. (An update is performed when an UPDATE command is entered or when an update is forced by the MERGE, RMERGE, EXCLUDE, INSERT, MOVE, RESEQ, COMPILER, EXECUTE, TYPE, REPLACE, UTILITY, or SAVE commands. The LIST command does not cause an update.)

A

The A option lists only altered lines in the workfile. Each line deleted by a DELETE or single-line entry is listed and flagged with a minus sign (-). Each line that has been modified by a FIX is listed and flagged with an "F". Each single-line entry that results in new text is flagged as follows: (1) for an old line replaced by a new line, the flag is an "R"; and (2) for a new line, the flag is "I". Blank lines are inserted to separate the listing into groups; a group contains a single replaced or fixed line (listed before and after) or an insertion or a deletion. Consecutively inserted or deleted lines are shown as a single group.

AF

The output from the AF option is the same as the A option output, except that fixed lines are shown as the line appeared before the FIX. The line showing the record before the FIX is flagged with a minus sign (-). (This option is functionally equivalent to the COMPARE option.)

AN

The output for the AN option is the same as the A option output, except that the neighboring lines of a new line are listed and are not flagged.

ANF

The ANF option combines the effects of the AN and AF options. Lines are shown before fixing, and the neighbors of new lines are shown. ANF and AFN are synonymous.

CHANGES

The output for the CHANGES option is the same as the A option output, except that only the sequence numbers (not the text) of deleted lines appear in the list, and groups of altered lines are not separated by vertical lines.

COMPARE

The COMPARE option is functionally equivalent to the AF option.

FLAG

The FLAG option causes each line in the workfile to be listed with the altered lines flagged as described in the A option description. Groups of altered lines are not separated by vertical spaces.

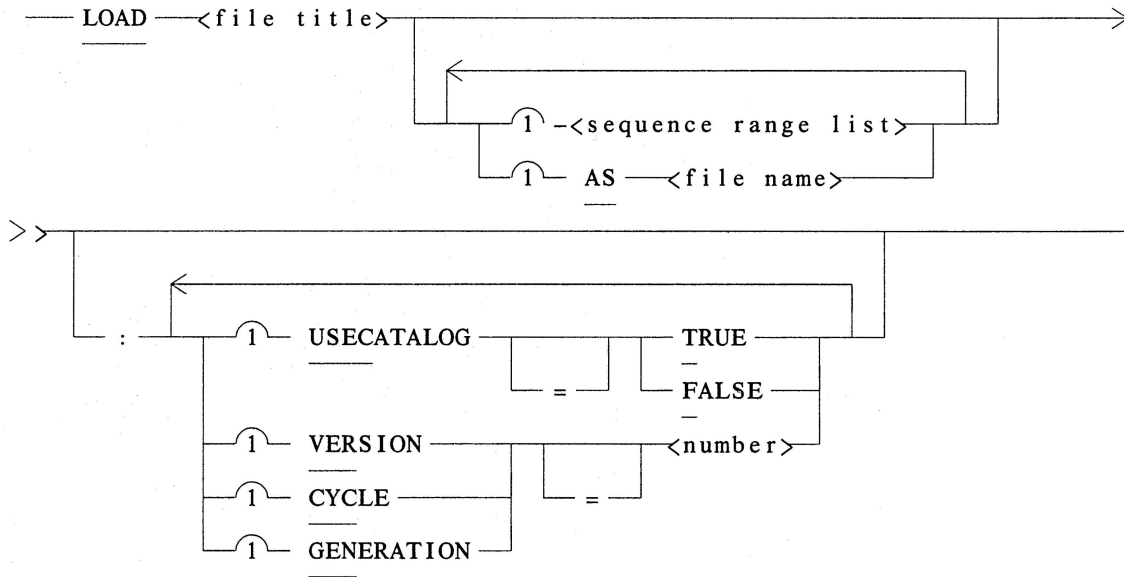
Examples

```
1
100 BEGIN
200 REAL Y;
300 Y:=2;
400 END.
#
```

```
LIST (UZER)FILE ON USERPACK 200-400 @72-80
#FILE (UZER)FILE ON USERPACK
200 THIS IS
300 COMMENT
400 SECTION
#
```

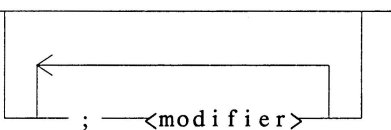
LOAD

Syntax



The LOAD and GET commands are synonyms. (Refer to the GET command in this section.)

LOG**Syntax**

— LOG — <SYSTEM/LOGANALYZER parameters> 

Semantics

The LOG command causes execution of the utility SYSTEM/LOGANALYZER. This utility extracts specific information as specified by <SYSTEM/LOGANALYZER parameters> from the system log file and displays the information at the user's terminal.

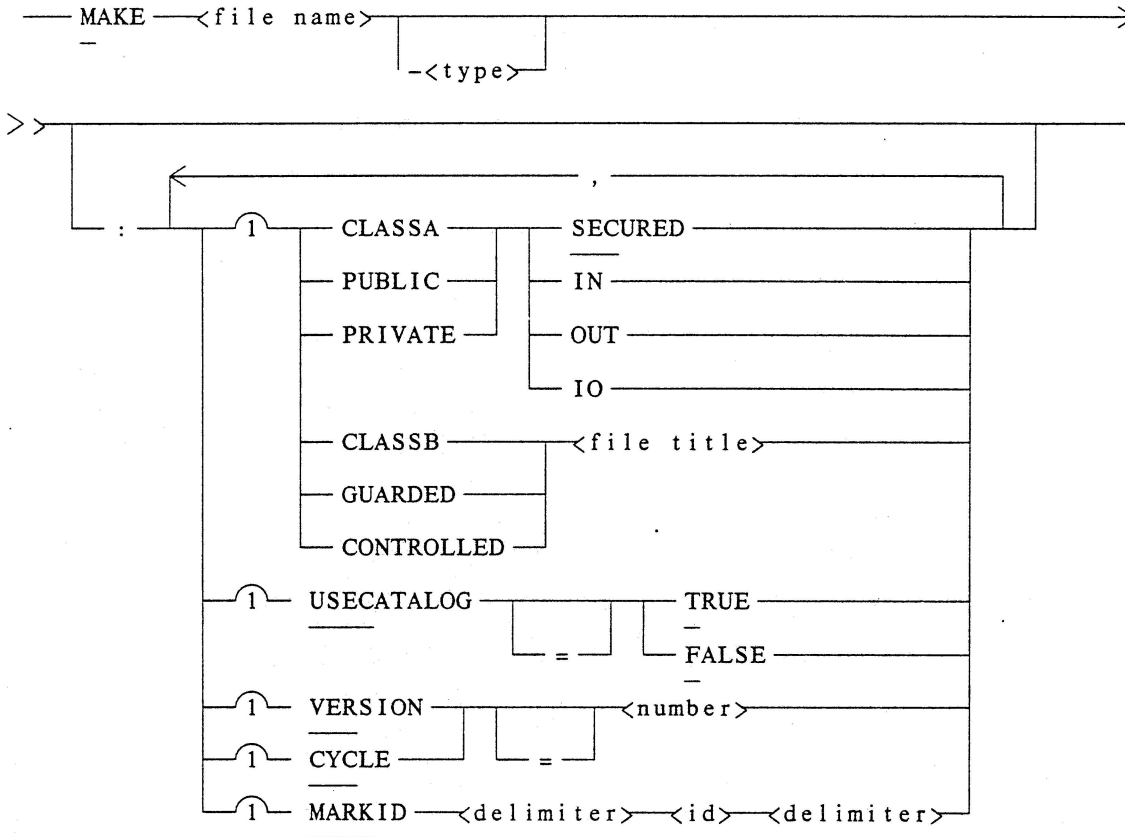
<SYSTEM/LOGANALYZER parameters> must be a string of standard options of the utility. The <SYSTEM/LOGANALYZER parameters> may be omitted, indicating the ALL option; however, this option produces necessarily voluminous output, and file equation to a suitable output device is recommended.

SYSTEM/LOGANALYZER is documented in Section 6 of the the B 5000/B 6000/B 7000 Series System Software Operational Guide, Volume 2, form 5011679.

Examples

```
LOG MIX 6483
#RUNNING 4195
3.0.90 LOGANALYZER 04/21/78, 08:51:52, MCP: SYSTEM/MCP30090 ON DISK.
    III.0.129, SYSTEM SERIAL: #277
REQUEST IS: MIX 6483
LOG FILE IS: *SYSTEM/SUMLOG ON USERFILES.
(LOG FILE IS 958 RECORDS LONG)
JAN 26, 1978
    15:21:03      BOJ   6578 "RC PK065".
                        QUEUE: 0
                        ORIGINATING UNIT: 0
                        PRIORITY: 100
    15:21:09      6578 FAMILY REBUILDING: PK065  TEMP
    15:21:10      6578 PK 065 UNIT RC-ED
    15:21:10      EOJ   6578 "RC PK065".
                        0.780 SEC CPU, 5.014 SECS IO
                        MEM INTEGRAL: CODE=0.000, DATA=19.064
                        AVERAGE CORE USAGE: CODE=0 DATA=3297
                        ELAPSED TIME: 00:00:07

NORMAL TERMINATION FOR LOGANALYZER PROGRAM.
#
log all;file line(kind=printer)
#RUNNING 4137
#
```

MAKE**Syntax****Semantics**

The MAKE command creates a new workfile. If an unsaved workfile already exists, the following error message is displayed:

```
#REMOVE OR SAVE WORKFILE
```

The workfile must be removed or saved before another workfile can be created.

<file name> must define a new file within the user's library and becomes the name associated with the workfile. If <type> is not specified, the workfile is type SEQ by default. Allowable <type>s and the associated attributes are defined in BASIC CONSTRUCTS, Section 3.

<file name> cannot reference an existing file in the user's library. CANDE rejects such requests with a message similar to:

```
#SOURCE AND OBJECT FILE ALREADY PRESENT
```

CANDE REFERENCE MANUAL

The security of the workfile may be specified via CLASSA, PUBLIC, PRIVATE, CLASSB, GUARDED, SECURED, IN, IO, or OUT. The default security is PRIVATE IO.

For those installations that use file cataloging, USECATALOG may be specified. The USECATALOG option may be used only when the system option CATALOGOK is set.

The MARKID <delimiter> <id> <delimiter> option establishes an initial markid. The MARKID command may be used at any time afterwards to establish a new markid.

The following commands display the correspondence between the ID field of a record and the current value of the markid, if the MARKID option is specified:

1. FIND on a workfile using the TEXT option.
2. RANGE on a workfile using the TEXT option.
3. LIST (or PRINT) on a workfile without the UNSEQUENCED or PUNCH options.
4. REPLACE using the TEXT option, before the record ID field is replaced.

The displays from the preceding commands include the following:

1. The sequence number.
2. An asterisk ("*") if the record id is the same as the mark id; otherwise, a blank (" ").
3. The text.

Examples

```
make it
#WORKFILE IT: SEQ
```

```
MAKE TOP/SECRET : PRIVATE,OUT
#WORKFILE TOP/SECRET: SEQ
```

```
MAKE CAT/FILE: USEC=TRUE,VERSION=5,CYCLE=11
#WORKFILE CAT/FILE: SEQ
```

```
MAKE GUARDED/PROGRAM COBOL: GUARDED FBI ON TRIAL
#WORKFILE GUARDED/PROGRAM: COBOL
```

```
mak testonly a
#WORKFILE TESTONLY: ALGOL
```

```
MAKE CANDE/EX ALGOL:MARKID "ORIG"
#WORKFILE CANDE/EX : ALGOL, MARKID "ORIG" "
```

```
100XXX
200YYY
```

```
LIST
100*XXX
200*YYY
#
```

```
SAVE
#UPDATING
#WORKSOURCE CANDE/EX SAVED
```

```
GET CANDE/EX : MARKID "CHGED"
#WORKFILE CANDE/EX: ALGOL, 2 RECORDS, SAVED, MARKID "CHGED"
```

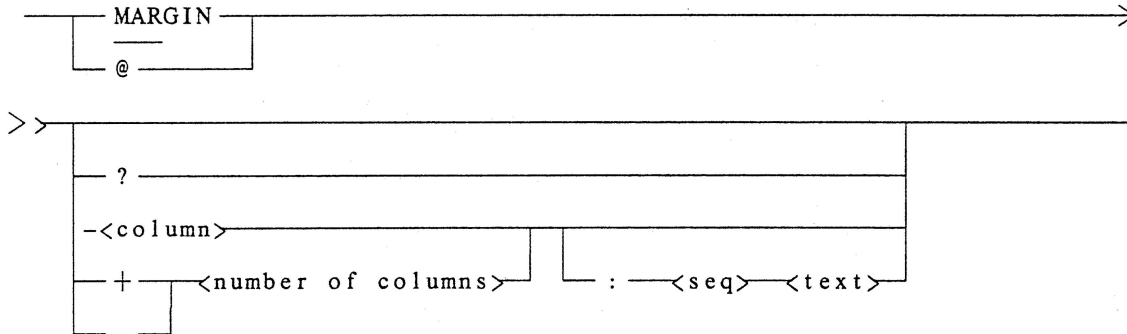
```
100XXXABC
```

```
LIST
100*XXXABC
200 YYY
#
```

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES. |

MARGIN

Syntax



Semantics

The MARGIN command controls the entry of text at the left margin of a line by inserting a specified number of blanks to the left of the text being entered. If the <column> option is specified, <column> represents an absolute column number where the text is to start. If the plus (+) or minus (-) <number of columns> option is specified, <number of columns> is relative to the previous margin setting.

If a question mark (?) is specified, the current margin setting is displayed.

If no <column>, question mark (?), or <number of columns> is specified, the margin is set to column 1.

If a colon (:) is specified, the margin specification applies only to the single-line entry that follows.

Margin specifications may be specified in automatic sequence mode by using the at sign (@) as the first character after the system-supplied sequence number. The margin specification applies only to the line on which it appears. The :<seq> option cannot be used.

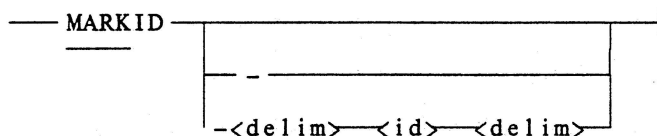
A line overflow may occur as a result of indentation. A notice is displayed, and the resulting line is truncated.

Examples

```
MAR 20 : 100 IF A > 99 THEN
@+4
#
MARGIN ?
#MARGIN @5
200%TEST COLUMN
L
100                IF A > 99 THEN
200      %TEST COLUMN
#
```

MARKID

Syntax



Semantics

The MARKID command can be used to place a value in the ID field of new or modified records. (Refer to Table 3-1 in this manual for a description of the ID field.)

If the MARKID command is specified, CANDE displays one of the following messages:

1. FILE TYPE CANNOT USE MARKID.

This display indicates that the FILETYPE does not have the ID field.

2. MARKID IS NOT IN USE.

This display indicates that no markid has been established for the workfile.

3. MARKID "<id>".

This display indicates that the value of the markid is the <id> plus as many blanks as are necessary to fill the ID field.

The MARKID <delimiter> <id> <delimiter> command replaces the old value of the markid (if any) with the new value specified by <id>. If this command is specified, CANDE displays one of the following messages:

1. FILE TYPE CANNOT USE MARKID

This display indicates that the FILETYPE does not have the ID field.

2. MARKID CAN ONLY BE <number> CHARACTERS LONG

The <id> was too long.

3. MARK "<id>"

This display indicates that the value of the markid is the <id> plus as many blanks as are necessary to fill the ID field.

CANDE REFERENCE MANUAL

The MARKID - command removes the markid from use. If this command is specified, CANDE displays one of the following messages:

1. MARKID IS NOT NOW IN USE

This display indicates that a markid was in use before the MARKID - command was entered.

2. MARKID IS NOT IN USE

This display indicates that no markid was in use before the MARKID - command was entered.

Examples

```

MAKE EXAMPLE ALGOL:MARK /FIRST/
#WORKFILE EXAMPLE: ALGOL, MARKID "FIRST"
100BEGIN
200 REAL I, K;
300 FILE REM(KIND=REMOTE);
400 READ(REM, /, I, K);
500 WRITE(REM, <"DIVIDE = " U>, I/K);
600END.
MARK /SECOND/
#UPDATING
#MARKID "SECOND"
FIX 400/R/WHILE NOT R
#
FIX 400/;/ DO
#
FIX 500/W/ W
#
REPLACE /K//J/:T
#WORKFILE EXAMPLE
#UPDATING
200 REAL I, J;
400* WHILE NOT READ(REM, /, I, J) DO
500* WRITE(REM, <"DIVIDE = " U>, I/J);
#
L
100 BEGIN
200* REAL I, J;
300 FILE REM(KIND=REMOTE);
400* WHILE NOT READ(REM, /, I, J) DO
500* WRITE(REM, <"DIVIDE = " U>, I/J);
600 END.
#
MARK-
"MARKID IS NOT NOW IN USE
L @1-90:U
BEGIN
REAL I, J;
FILE REM(KIND=REMOTE);
WHILE NOT READ(REM, /, I, J) DO
WRITE(REM, <"DIVIDE = " U>, I/J);
END.
#
MARK
#MARKID IS NOT IN USE
GET SYM/MINE:MARK .81 MAY 8.
#WORKFILE SYM/MINE: ALGOL, 756 RECORDS, SAVED, MARKID "81 MAY 8"

```

```

00000100FIRST
00000200SECOND
00000300FIRST
00000400SECOND
00000500SECOND
00000600FIRST

```

CANDE REFERENCE MANUAL

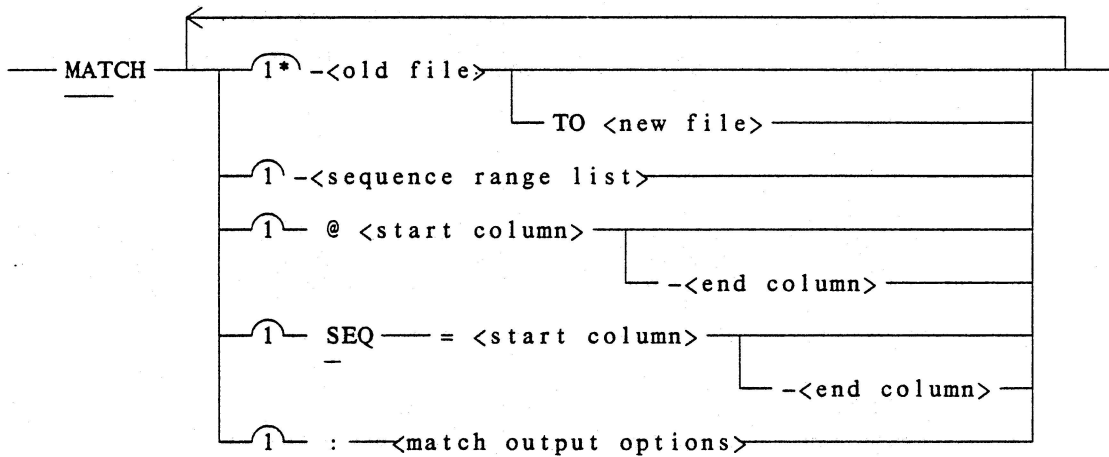
FIX 59450/BEGON/BEGIN

61400 REAL
MARKID .81 MAY 11.
#UPDATING
#MARKID "81 MAY 11 "
61470 RECCS,
MARK
#MARKID "81 MAY 11 "

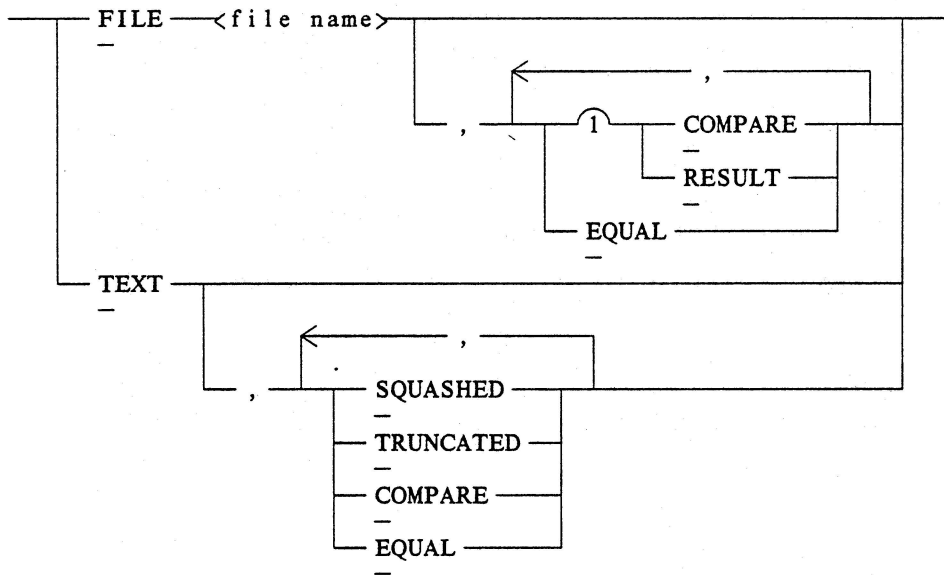
THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES. |

MATCH

Syntax



<match output options>



Semantics

The MATCH command compares two files and determines the differences between them. Output indicating the result of the comparison may be directed to the terminal or to a new file. The <old file>, or portions thereof, is compared with the <new file>; if a <new file> is not specified, the workfile is used by default.

For sequenced files, if a record is encountered in each file having the same sequence number, a character-by-character comparison of the records is done. If the two records are found to differ, the <new file> record is output as a replacement line. If a record exists at a sequence number in the <old file> but does not exist in the <new file>, that record is output as a deleted line. If a record exists at a sequence number in the <new file> but not in the <old file>, that record is output as an inserted line.

For unsequenced files, the first record of the <old file> is compared character-for-character with the first record of the <new file>, the second record is compared with the second record, and so forth.

Any two files that are MATCHed must be of the same type.

If a <sequence range list> is specified, the files are only compared within that sequence range. If a column specification is used (for example, @ <start column>), the records are compared within that column range. If the SEQ option is used, the indicated column range is used as sequence numbers for the comparison. The width of this sequence number cannot exceed eight digits, and the option may only be used with DATA files.

The results of the comparison are formed according to the output options used. By default, output is directed to the terminal and lists only the sequence numbers. Line deletions are preceded by a minus sign (-), line replacements by an "R", and line insertions by an "I".

The two available output options are described in the following paragraphs.

TEXT

The TEXT option directs output to the terminal; line deletions are preceded by a minus sign (-), replacements by an "R", and insertions by an "I".

If the COMPARE option is specified, the results are similar to the results of the COMPARE option in the LIST command.

If the EQUAL option is used, records that are the same are listed.

If the SQUASHED option is used, any group of multiple blanks is reduced to a single blank for output of the record.

If the TRUNCATED option is used, a line too long for the terminal is truncated to one terminal line.

FILE <file name>

The FILE <file name> option directs output to the specified file. The contents of the file are the same as those given for TEXT. The resulting file is of type DATA. Files created with the MATCH command are unconditionally crunched.

If the COMPARE option is used, the contents of the file are the same as the results given for the use of COMPARE with the TEXT option.

If the RESULT option is used, the result file is the same type as that of the files MATCHed.

CANDE REFERENCE MANUAL

For language source files, line insertions and replacements are put in the result file. Single-line deletions are replaced by a dollar sign (\$) record. Multiple-line deletions in sequence cause output of a \$SET VOIDT record at the starting sequence number and a \$POP VOIDT at the ending sequence number.

For DATA file comparisons using the sequence number column specification, line replacements and insertions are put in the result file. For deletions, a blank record is put in the file with sequence number intact.

For DATA file comparisons not using the sequence number column specification, when records differ, the <new file> record is placed in the result file. If the <new file> is longer than the <old file>, trailing unmatched records are placed in the output file. If the <old file> is longer than the <new file>, the trailing unmatched records are not placed in the output file.

If the EQUAL option is used, records that are the same are placed in the file.

Examples

```
L
100 THIS IS IN BOTH FILES
200 DIFFERENT COLUMNS
300 SAME COLUMNS
400 NEW FILE
#
```

```
LIST A/B
#FILE (UZER)A/B ON USERPACK
100 THIS IS IN BOTH FILES
 150 OLD FILE
200 DIFFERENT COLUMNS
300 SAME COLUMNS
#
```

```
MATCH A/B:T,C
#UPDATING
#MATCHING A/B TO WORKFILE
```

```
-150 OLD FILE
```

```
-200 DIFFERENT COLUMNS
R200 DIFFERENT COLUMNS
```

```
 300 SAME COLUMNS
I400 NEW FILE
#
```

```
MATCH A/B TO WORKFILE @15-20
#MATCHING A/B TO WORKFILE
-150,R200,I400
```

MCS**Syntax**

 MCS <file title> _____|

Semantics

The MCS command causes control of the station to be transferred to another Message Control System (MCS). <file title> must designate an NDL-declared MCS name. If an unsaved workfile exists, the following message is displayed:

```
#REMOVE OR SAVE WORKFILE
```

The unsaved workfile must either be removed or saved.

This command is the same as the MCS control command (?MCS - refer to Section 5) except that this form of the command may be queued if the station is busy, and the control command may not. The user must be logged-on to use the noncontrol command form.

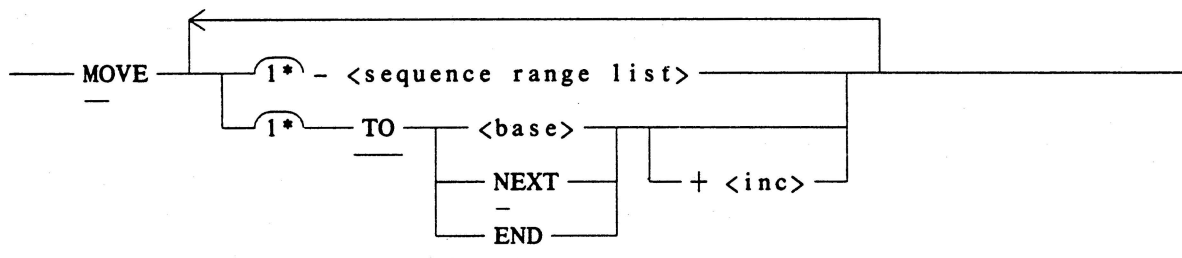
The user is logged-off and the session terminated before control of the station is given to the desired MCS.

Examples

```
MCS SYSTEM/DIAGNOSTICMCS ON UZERPACK
#END SESSION 4241 ET=58:46.2 PT =1.8 IO=0.3
#USER = UZER 10:07:55 04/21/78
SYSTEM/DIAGNOSTICMCS(3.0.0) NIF:ELMONTENDL(2.9.0) YOU ARE T1(LSN:1)
```

```
MCS SYSTEM/TIONDL ON PACK
#END SESSION 4245 ET=24:30.0 PT=0.7 IO=0.1
#USER = UZER 10:35:50 04/21/78
[ TIOMCS ] == WELCOME == YOU ARE TTY0 (LSN 13) SESSION 4247
```


CANDE REFERENCE MANUAL

MOVE**Syntax****Semantics**

The MOVE command moves lines from one place to another within the workfile. The lines are moved as a contiguous block and are assigned new sequence numbers. Lines are moved, not copied; lines in the <sequence range list> no longer reside at their old numbers. (To copy lines in the workfile, the INSERT command is used.)

<sequence range list> specifies the sequence range(s) of the lines to be moved within the workfile.

The new sequence numbers of moved lines are determined by assigning an initial value to the first line and incrementing that value for each succeeding line. The initial value must be specified in one of three ways:

1. <base> specifies the value explicitly. <base> must be an <integer>.
2. NEXT sets the initial value to the next sequence number that would have resulted from the most recent MOVE, INSERT, RESEQ, or SEQ command. A default value of 100 is used if no such command has appeared since the last MAKE, GET, or DELETE ALL command.
3. END sets the initial value to the largest sequence number currently in the file, plus the current specified (or default) increment. <inc> defines the increment value to use for successive sequence numbers in the block. <inc> must be an <integer>. If <inc> is unspecified, the increment value from the most recent MOVE, INSERT, RESEQ, or SEQ command is used (100 is used if none of these commands have appeared).

The range of new sequence numbers cannot overlap any lines already in the file, nor can the numbers exceed the largest sequence number that may be expressed in the sequence number field.

Examples

```
GET TESTONLY;LIST
#WORKFILE TESTONLY: SEQ, 15 RECORDS, SAVED
100 LINE 1
200 LINE 2
300 LINE 3
400 LINE 4
500 LINE 5
600 LINE 6
700 LINE 7
800 LINE 8
900 LINE 9
1000 LINE 10
1100 LINE 11
1200 LINE 12
1300 LINE 13
1400 LINE 14
1500 LINE 15
#
MOVE 1000-1300 TO 10+1
#UPDATING
#
L
10 LINE 10
11 LINE 11
12 LINE 12
13 LINE 13
100 LINE 1
200 LINE 2
300 LINE 3
400 LINE 4
500 LINE 5
600 LINE 6
700 LINE 7
800 LINE 8
900 LINE 9
1400 LINE 14
1500 LINE 15
#
MOVE 1400,1500 TO NEXT
#UPDATING
#
L
10 LINE 10
11 LINE 11
12 LINE 12
13 LINE 13
14 LINE 14
15 LINE 15
100 LINE 1
200 LINE 2
300 LINE 3
400 LINE 4
500 LINE 5
600 LINE 6
700 LINE 7
800 LINE 8
900 LINE 9
#
```

NEWS

Syntax

NEWS _____|

Semantics

The NEWS command causes the contents of the CANDE news file to be listed on the user's terminal. If no news file exists, CANDE responds with the following display:

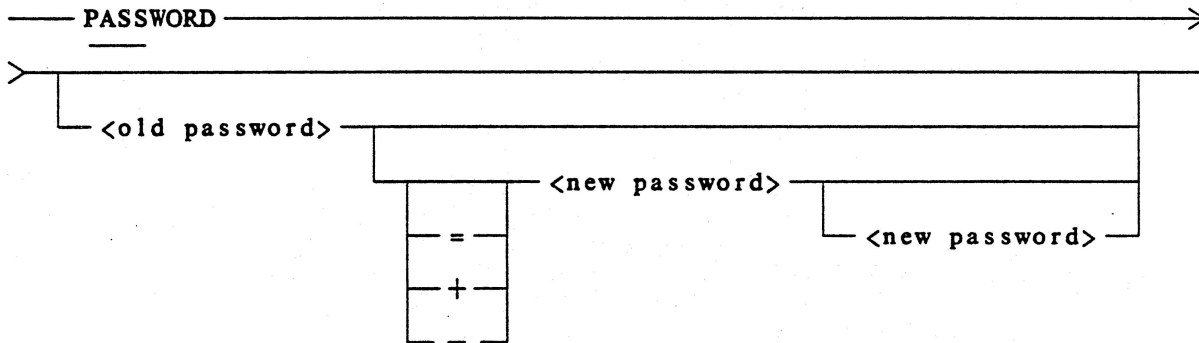
```
#NO NEWS
```

Example

```
NEWS  
COMING UP ON NEW RELEASE NEXT WEEK  
#
```

PASSWORD

Syntax



Semantics

The PASSWORD command, in its simplest form, allows a password to be changed for a user. This command is also used to alter a list of passwords if more than one password is allowed for the usercode.

To change passwords, all or part of the required information can be entered with the PASSWORD command. Any information not initially provided is requested by CANDE. For verification purposes, the <new password> must be entered twice.

In the simplest case, a user is allowed only one password. The only form of the PASSWORD command that is valid for this case is as follows:

```
PASSWORD <old password> <new password> <new password>
```

This form changes the user's password from the current password (<old password>) to a new password (<new password>).

In the case where a usercode is allowed more than one password, passwords in the list can be changed or deleted by using the equal (=) or minus (-) operators, respectively. Passwords can be added to the list by using the plus (+) operator.

Pragmatics

In each of the cases below, FIRSTONE must be the first password in the password list (or its successor if FIRSTONE was explicitly changed, as is done in the last example shown). The first password in the list is the first password defined for that usercode.

To add PW to the password list, the following command is entered:

```
PASSWORD FIRSTONE + PW PW
```

To delete PW from the password list, the following command is entered:

```
PASSWORD FIRSTONE - PW PW
```

To replace the whole password list with the single entry PW, the following command is entered:

```
PASSWORD FIRSTONE = PW PW
```

To remove the requirement for a password at log-in, the following command is entered:

```
PASSWORD FIRSTONE - FIRSTONE FIRSTONE
```

To replace the password OLDPW by PW, the following command is entered:

```
PASSWORD OLDPW PW PW
```

The operations for password lists are constrained by the MINPW and MAXPW bounds established by the installation (in the USERDATAFILE) for the number of passwords that can be defined for any given usercode.

Occasionally, the USERDATAFILE is frozen so that no changes can be made. If any attempt to change a password (or password list) is made during this time, CANDE rejects the attempt with the following message:

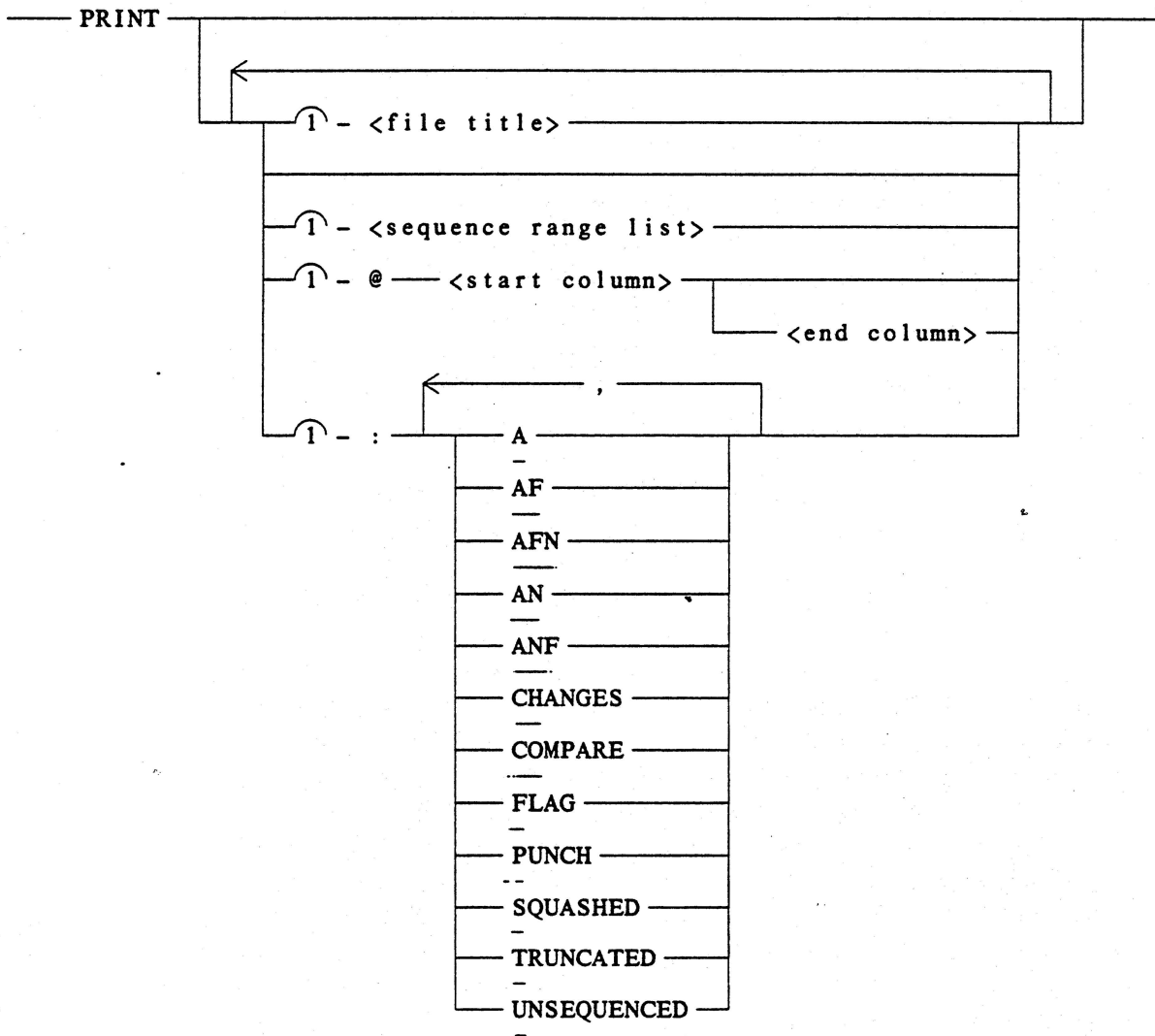
```
#USERDATAFILE FROZEN; TRY LATER.
```

Examples

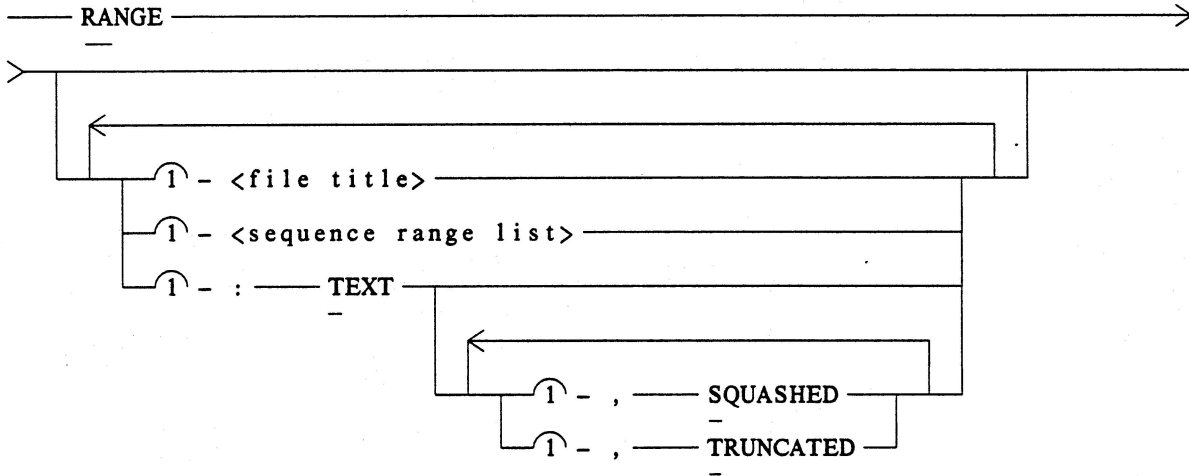
```
PASSWORD
#ENTER CURRENT PASSWORD, PLEASE.
JOSEPH
#ENTER NEW PASSWORD PLEASE.
JOE
#RE-ENTER NEW PASSWORD PLEASE.
JOE
#
```

PRINT

Syntax



The PRINT and LIST commands are synonyms. (Refer to the LIST command in this section.)

RANGE**Syntax****Semantics**

The RANGE command displays the number of lines within one or more sequence ranges of a file (by default, the workfile). The lowest and highest sequence numbers in the sequence range(s) are also displayed.

<file title> specifies the file to be ranged; this file can be any file to which the user is allowed access. If unspecified, the workfile is assumed.

<sequence range list> limits the ranging to particular areas of the file; the default is the entire file. If a <sequence range> in the <sequence range list> consists of a single sequence number or the single keyword END, the sequence numbers (and optionally, the text) of the preceding and succeeding lines are also displayed.

The TEXT option causes entire lines to be printed in the range output, instead of sequence numbers. The two output format options available are TRUNCATED and SQUASHED; these options affect the text as described below.

TRUNCATED

If the text portion of a record does not fit in the available line width of the terminal, the record is truncated.

SQUASHED

Any group of multiple blanks is reduced to a single blank.

Examples

```
LIST
100 one
200 two
300 three
400 four
500 five
600 six
700 seven
800 eight
900 nine
1000 ten
#
RANGE
# 10 RECORDS: 100 THRU 1000
#
RANGE:TEXT
# 10 RECORDS:
100 one
  THRU
1000 ten
#
RANGE 500
400, 500, 600
#
RANGE 500:TEXT
400 four
500 five
600 six
#
RA 100-300,400-END,1025-1050
# 3 RECORDS: 100 THRU 300

# 7 RECORDS: 400 THRU 1000

#NO RECORDS IN 1025-1050
#
RA 550:TEXT
500 five
600 six
#
```

RECOVER

Syntax

```
— RECOVER —————|  
  |  
  | <recovery number> |  
  |
```

Semantics

The RECOVER command recalls a recovery file as the workfile. Recovery of a recovery file is achieved by specifying the desired <recovery number>. If a <recovery number> is not specified, CANDE provides a list of all recovery files in the following format:

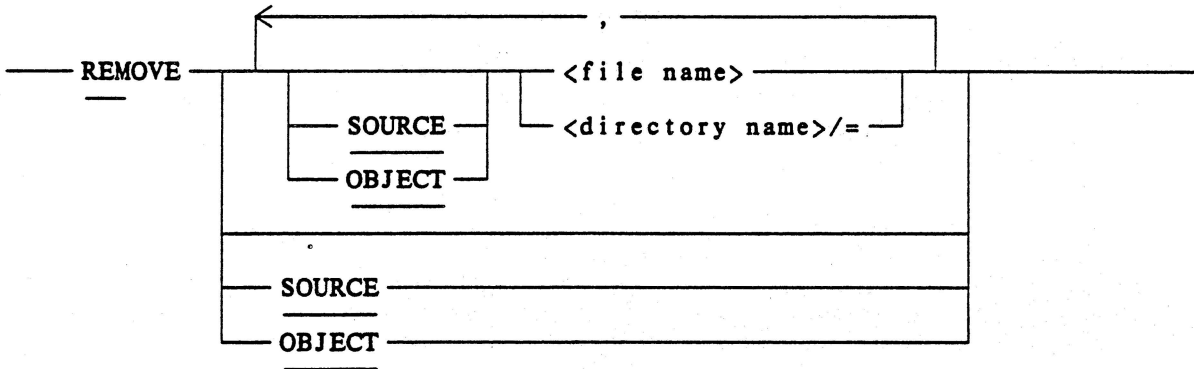
```
#RECOVERY DATA:
```

```
<recovery number> <workfilename> .( <date> )
```

```
.  
.  
.
```

Examples

```
rec  
#RECOVERY DATA ON USERPACK  
  100 ZOT (04/21/78)  
#  
recover 100  
#WORKFILE ZOT: SEQ
```

REMOVE**Syntax****Semantics**

The REMOVE command is used to remove the workfile, library files, or directories (and the subordinate files of the directory) from the user's library.

Use of the SOURCE or OBJECT <file name>s must be quoted to prevent ambiguity with the keywords SOURCE and OBJECT.

When a file is removed, it immediately disappears from the system directory; therefore, no program can find it. The actual data areas remain available to any program that already had that file open at the time of the removal. CANDE saves a copy of a file, if necessary, in the workfile area, if a file being removed is required for the workfile.

The semantics of each syntactic variation of the syntax diagram are described subsequently under headings reflecting the syntactic form.

REMOVE

The REMOVE form removes the workfile and any associated object file. If SOURCE is specified, the workfile is removed, but any associated object file remains available to be executed or saved. If OBJECT is specified, any object file associated with the workfile is removed, and the workfile (source) remains.

REMOVE <file name>

The REMOVE <file name> removes any file with the specified name which exists in the user's library. Any file named OBJECT/<file name> is also removed. If SOURCE is specified, the action is the same except OBJECT/<file name> is not removed. If OBJECT is specified, only OBJECT/<file name> is removed.

CANDE REFERENCE MANUAL

REMOVE <directory name>

The REMOVE <directory name> form causes any subordinate files of the directory to be removed if any directory with the specified <directory name> exists in the user's library. The files of OBJECT/<directory name> are also removed, if they exist. If SOURCE is specified, the action is the same except that the files of OBJECT/<directory name> are not removed. If OBJECT is specified, only the files of OBJECT/<directory name> are removed.

Examples

```
G D/F
#WORKSOURCE D/F: SEQ, 1 RECORD, SAVED
FIX 100/2/1
#
REM D/=,F/=,TEMP,SOURCE UTIL, Z
#WORKSOURCE D/F RETRIEVED
#1 FILE IN (UZER)D/= REMOVED ON USERPACK
#1 FILE IN (UZER)F/= REMOVED ON USERPACK
#(UZER)TEMP ON USERPACK REMOVED
#(UZER)UTIL ON USERPACK REMOVED
#NO FILE (UZER)Z ON USERPACK
FILES D
#NO FILE(S) ON USERPACK
WHAT
#WORKFILE D/F: SEQ
REM
#
```

RENEW**Syntax**

— RENEW — <sequence range list> —————|

Semantics

The RENEW command allows portions of the workfile to be renewed to the condition in which they existed the last time the workfile was updated. The <sequence range>s specified need not be in ascending order.

Examples

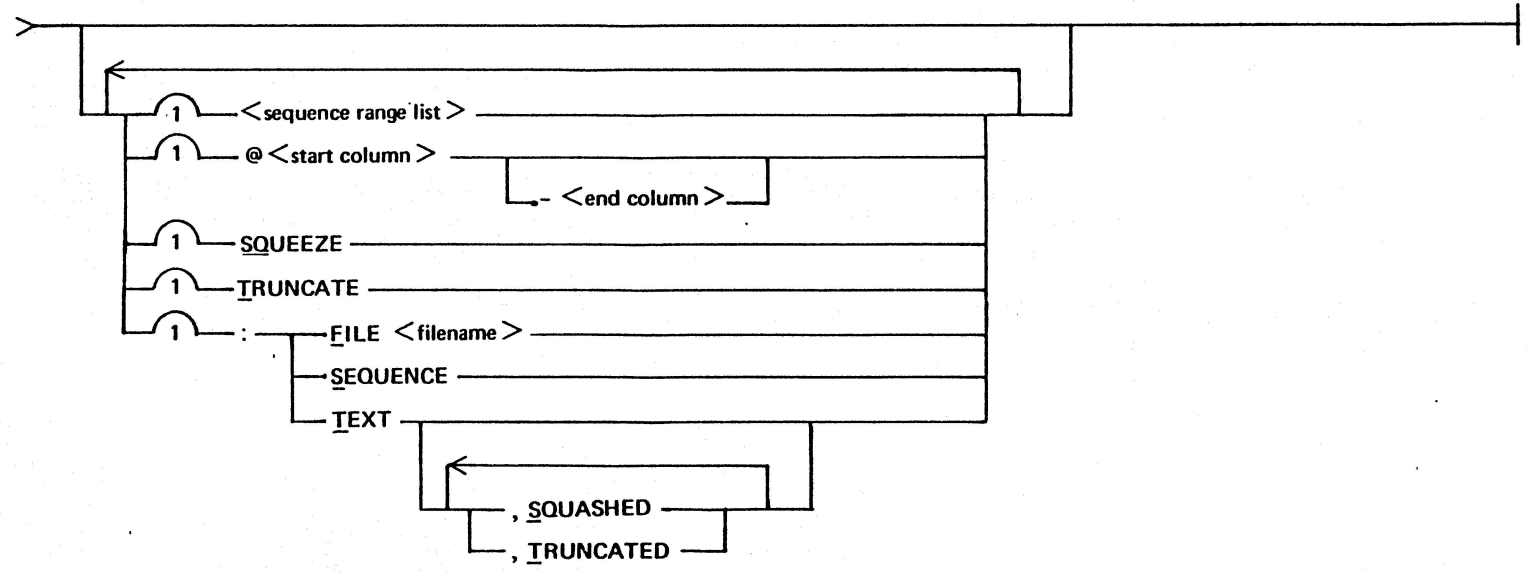
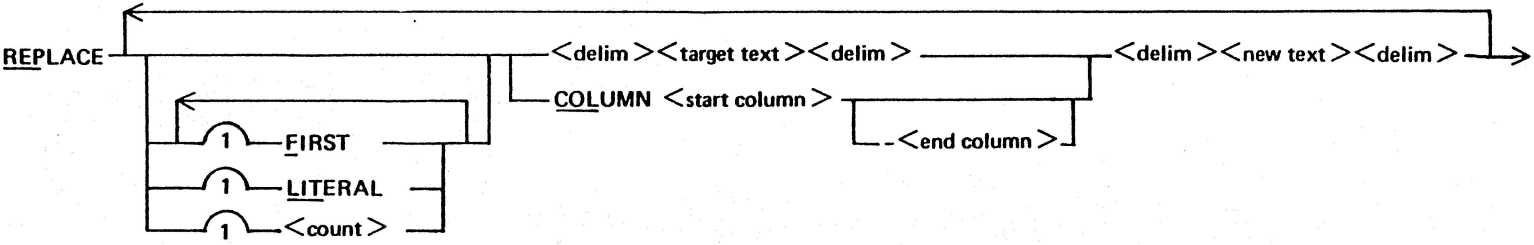
```

GET X
#WORKFILE X: SEQ, 3 RECORDS, SAVED
LIST
2000 THE RAIN IN SPAIN STAYS  MAINLY
4000 IN THE PLAIN.
6000 THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
#
FIX 4000/IN/ON
#
8000 THE EARLY BIRD GETS THE WORM.
DELETE 6000
#
LIST
2000 THE RAIN IN SPAIN STAYS  MAINLY
4000 ON THE PLAIN.
8000 THE EARLY BIRD GETS THE WORM.
#
RENEW 0-10000
#
LIST
2000 THE RAIN IN SPAIN STAYS  MAINLY
4000 IN THE PLAIN.
6000 THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
#

```

REPLACE

Syntax



Semantics

The REPLACE command searches all, or part, of the workfile for occurrences of specific target text and replaces the target text with new text. The system responds with a pound sign (#) to signify completion.

The LITERAL option specifies that the search mode for the target text is to be the literal mode. The default search mode is token mode. (Refer to the Pragmatics heading under the FIND command in this section, where search modes are described.)

<count>, if specified, must be an unsigned <integer>. <count> limits the maximum number of replacements to be performed. Any appearances of the target text after <count> occurrences are ignored.

If FIRST is specified, each line is only searched for the first occurrence of the target text. Any occurrences later in the line are ignored.

<dlm> may be any special character except comma (,), colon (:), or the at sign(@).

The target text for a replace must be specified in one of the three forms described in the following paragraphs.

<target text>

<target text> explicitly defines the target text and must be bounded by two matching delimiters <dlm>.

COLUMN <start column> - <end column>

The form COLUMN <start column> - <end column> defines the target text to be all text on each line between the character positions specified by <start column> and <end column>, inclusive.

COLUMN <start column>

The COLUMN <start column> form specifies that the new text (<new text>) is to be inserted beginning at character position <start column> of each line. The remainder of each line is shifted right to accommodate the <new text>.

Adjustment for different lengths of target text and new text is made by shifting the right side of the line (or column range specified by @<start column> - <end column>) to the left or right, deleting or adding trailing blanks as required. A line overflow error is detected whenever the adjustment would shift nonblank characters off the end of the line [(or the (@) column range)]. By default, the replacement is not performed if an overflow occurs, and a #<seq>-SKIPPED. message is displayed.

The two options available for alternate treatment of overflow conditions are described in the following paragraphs.

SQUEEZE

The SQUEEZE option causes strings of multiple blanks to be shortened (if possible) to make room for the substitution. If sufficient squeezing is impossible, the line is left "unsqueezed", the substitution is skipped, and a #<seq>-SKIPPED. message is displayed.

CANDE REFERENCE MANUAL

TRUNCATE

The TRUNCATE option causes the characters shifted off the right side of a line overflow to be lost.

<sequence range list> limits the REPLACE operation to one or more <sequence range>s in the workfile.

The three output options available are described in the following paragraphs:

FILE <file name>

The FILE <file name> option causes each line modified by the REPLACE to be output to a file in the user's library. <file name> specifies the title of that file. <file name> need not be a new file name in the user's library. However, any existing file titled <file name> is first removed before the new file is created.

Files created by the REPLACE command are crunched by default. If a crunched file is not desired, the output option NOCRUNCH may be specified.

SEQUENCE

The SEQUENCE option causes the sequence number of each line modified by the REPLACE to be displayed at the terminal. Asterisks in this output flag lines where more than one replacement occurred.

TEXT

The TEXT option causes each line modified by the REPLACE to be displayed at the terminal in the modified form. The SQUASHED and TRUNCATED options apply to the terminal output only and do not affect the line images in the workfile. SQUASHED causes a sequence of blanks to be printed as a single blank. TRUNCATED causes the output line to be truncated to the character width of the terminal, if necessary. By default, lines that exceed terminal width are split across two or more lines.

Examples

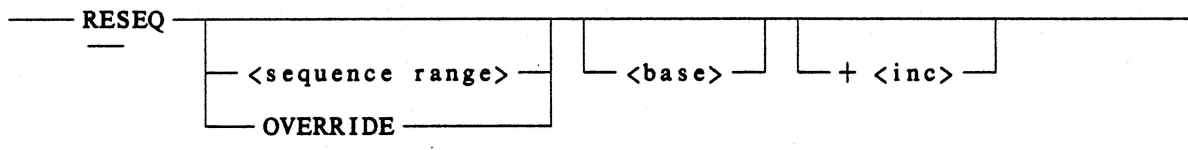
```
G FRUIT
#WORKFILE FRUIT: SEQ, 5 RECORDS, SAVED
LIST
100 APPLE ORANGE PEAR
200 GRAPE PLUM APPLE
300 CHERRY APPLE LEMON
400 LIME PEAR BANANA
500 APPLE ORANGE APPLE
#
REPLACE /APPLE/ /LEMON/ :S
#WORKFILE FRUIT
#UPDATING
100, 200, 300,*500
#
LIST
100 LEMON ORANGE PEAR
200 GRAPE PLUM LEMON
300 CHERRY LEMON LEMON
400 LIME PEAR BANANA
500 LEMON ORANGE LEMON
#
REPLACE .X..BITWORD. :FILE REVISED,NOCR SQUEEZE
```

CANDE REFERENCE MANUAL

#WORKFILE FRUIT
#UPDATING
#

RESEQ

Syntax



Semantics

The RESEQ command assigns new sequence numbers to lines in the workfile without changing the order of appearance of any lines.

An initial value is assigned as the sequence number of the first line to be resequenced. The value is incremented for each subsequent line.

If a range of sequence numbers is specified, only the specified part of the workfile is resequenced. By default, the entire workfile is resequenced.

An initial value may be specified as the integer <base>; otherwise, the starting value of the sequence range is used. <inc> may be specified as the increment for successive new sequence numbers. If no <inc> appears, the increment from the most recent MOVE, INSERT, RESEQ, or SEQ command is used. The default of 100 is assumed if no such command has occurred since the last MAKE, GET, or DELETE ALL command.

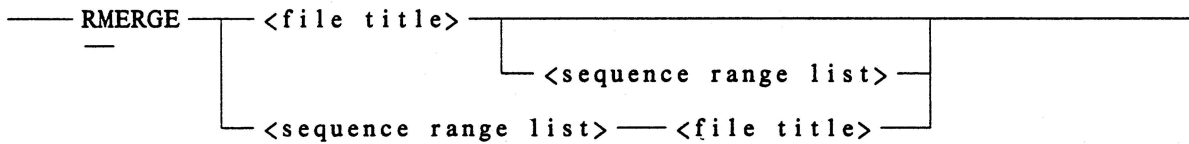
If a <sequence range> is specified, the new sequence numbers must fall within those bounds. RESEQ does not change the order of appearance of the lines.

OVERRIDE causes the sequence numbers currently in the file to be ignored. This option may be used only with a complete file that has no changes pending. The expected application is GET <file title>; RESEQ OVERRIDE, thereby sequencing a file that was previously unacceptable to CANDE because of sequence errors.

CANDE REFERENCE MANUAL

Examples

```
list
5 FIRST LINE
30 SECOND
72 THIRD
73 FOURTH
150 FIFTH
1000 SIXTH
#
reseq
#UPDATING
#
1
100 FIRST LINE
200 SECOND
300 THIRD
400 FOURTH
500 FIFTH
600 SIXTH
#
reseq 123-456 +5
#UPDATING
#
1
100 FIRST LINE
123 SECOND
128 THIRD
133 FOURTH
500 FIFTH
600 SIXTH
#
```

RMERGE**Syntax****Semantics**

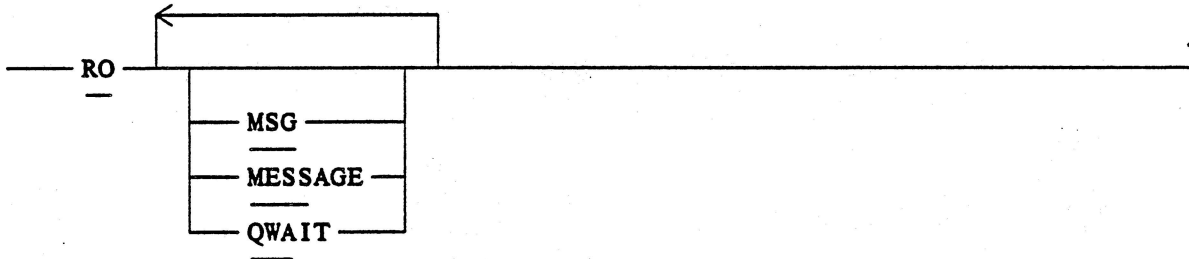
The RMERGE command copies a file (<file title>) into the workfile, collating the sequence numbers from both files and preserving the <file title> records wherever matching sequence numbers occur. A pound sign (#) acknowledges completion of the RMERGE operation.

The <sequence range list> option limits the RMERGE to specific portions of <file title>; the default is the entire file.

The RMERGE command performs the same function as MERGE, except where sequence numbers coincide; in that case, the <file title>, rather than the workfile version, takes precedence.

Examples

```
list
100 workfile - line 1
200 workfile - line 2
300 workfile - line 3
400 workfile - line 4
500 workfile - line 5
600 workfile - line 6
#
list rmergefile
#FILE (USERID)RMERGEFILE ON PACKID
100 rmergefile - line 1
150 rmergefile - line 2
200 rmergefile - line 3
250 rmergefile - line 4
#
rmerge rmergefile
#UPDATING
#
list
100 rmergefile - line 1
150 rmergefile - line 2
200 rmergefile - line 3
250 rmergefile - line 4
300 workfile - line 3
400 workfile - line 4
500 workfile - line 5
600 workfile - line 6
#
```

RO**Syntax****Semantics**

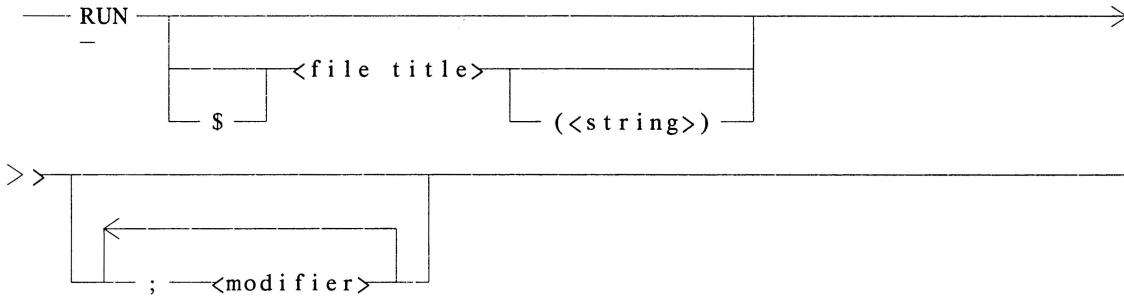
The SO and RO commands control the setting and resetting of station options. SO sets station options, and RO resets station options. The current state of station options is displayed if either the RO or SO command is entered without a following station option name.

MESSAGE and MSG are synonyms. These options control the automatic display of job- and task-related messages at the terminal during a CANDE session. If MSG is set, all job- and task-related messages are automatically displayed. When MSG is reset, only RSVP and DS messages are displayed on the terminal.

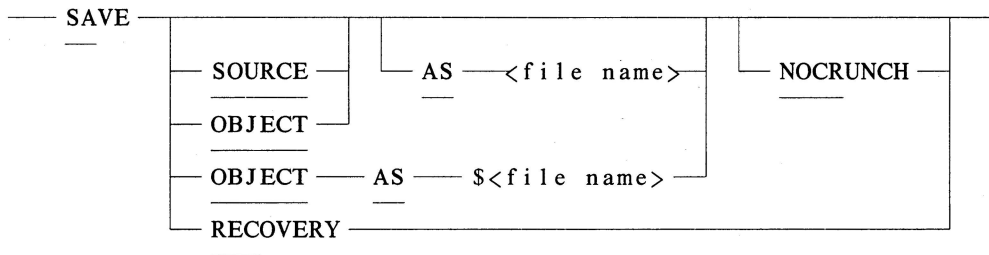
The QWAIT option specifies that the state of the user's queued input is to be set to waiting, rather than pending, in the event of an error. (Refer to Queue Manipulation in Section 2 for details.)

Examples

```
RO
#MESSAGES SET, QWAIT SET
ro messages
#MESSAGES RESET
RO QWAIT
#QWAIT RESET
RO
#MESSAGES RESET, QWAIT RESET
```

RUN**Syntax**

The RUN and EXECUTE commands are synonyms. (Refer to the EXECUTE command in this section.)

SAVE**Syntax****Semantics**

The SAVE command causes the current workfile and/or its associated object file to be saved in the user's library. The workfile is saved in the user's library using the current workfile name (OBJECT/workfilename for object files). Any existing file(s) of the same name are removed.

Source files may be copied under several different names; object files cannot be saved under more than one name.

After a workfile has been saved, it is still available for further editing. The recently saved file serves as the file part of the workfile; therefore, it remains present and unmodified until the next UPDATE.

An object file saved (but not saved AS) remains associated with the workfile and is available to be RUN or EXECUTED until any changes to the workfile are entered.

A SAVE AS <file name> saves the workfile in the user's library using the specified <file name>. If an associated object file exists, it is saved as OBJECT/<file name>. A SAVE SOURCE AS <file name> or SAVE OBJECT AS <file name> saves only the worksource or workobject, respectively. A SAVE OBJECT AS \$<file name> saves the workobject as the <file name> without an OBJECT/ preceding. If <file name> exists, the SAVE is not performed.

If NOCRUNCH is specified, the files are not crunched when they are saved. The default is to crunch all files.

Pragmatics

The use of the SAVE RECOVERY is a convenient way of temporarily saving files. Because CANDE does not update in this case, larger files are stored more quickly than with a SAVE <file id>. (Refer to the RECOVER command in this section for file retrieval.)

CANDE REFERENCE MANUAL

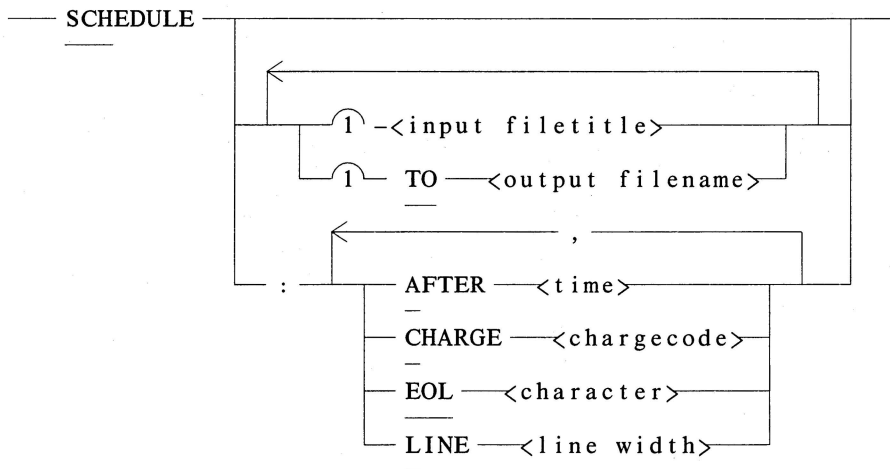
Examples

```
WHATS
#WORKFILE A:ALGOL
SAVE
#UPDATING
#WORKSOURCE SAVED
```

```
WHATS
#WORKFILE TEST:ALGOL
SAVE SOURCE AS NEW/PROG
#UPDATING
#WORKSOURCE TEST SAVED AS (UZER)NEW/PROG ON USERPACK
C
#COMPILING 5084
#ET=18.8 PT=0.5 IO=0.4
SA OBJECT AS $TEST
#WORKOBJECT TEST SAVED AS (UZER)TEST ON USERPACK
```

SCHEDULE

Syntax



Semantics

With a SCHEDULE command, the user presents a file of CANDE commands and program input. CANDE performs these commands in a separate schedule session that is independent of the user's station. The schedule session may begin immediately or after a specified time. The schedule input is merged with any session output to generate a schedule output file, which is an image of the information that would have appeared on a terminal had the session been interactive. The user may inquire about schedule sessions with a STATUS command and may terminate a session with a STOP command.

The <input filetype>, if present, specifies the file containing the schedule-session input and may specify any file accessible to the user. If no name is specified, the workfile is used. CANDE copies the input into a special schedule file, so the named file or workfile may be changed or removed as soon as the SCHEDULE command is completed (with no effect on the scheduled session). The input file may be of any type (FILEKIND) recognized by CANDE. Only the text portion of each line is processed by CANDE or user programs as input; the text field cannot exceed 84 characters in width.

The <output filename>, if present, names the file to receive the session terminal image; it may have no (usercode) or (*) prefix. If the <output filename> is omitted, a name is generated by prefacing SCHOUT/ to the <input filetype> or workfile name, respectively. Any usercode prefix or family suffix in the <input filetype> is ignored. The output file name, whether explicit or default, must be unique from any file already in the user's library. The output file is created with the security set to PRIVATE when the SCHEDULE command is processed. Initially, the output file contains a single line, #SCHEDULE #00017 IS SCHEDULED, followed by a blank line. These lines are overwritten when the scheduled session begins. The output file is of type SCHED (FILEKIND=SCHEDULEFILE); because it contains no sequence numbers, CANDE LISTS it with none. Any other CANDE command (such as GET, FIND, INSERT, WRITE) treats a SCHED file as type CDATE.

CANDE REFERENCE MANUAL

By default, a scheduled session is immediately considered for processing, subject only to availability of resources. The AFTER <time> option may be used to defer processing until after a specified time of the day. The <time> is specified as an integer in 24-hour notation; thus, 0 represents midnight (1, ..., 2359 represent 12:01 a.m., ..., 11:59 p.m.). Any <time> value less than the current time of day refers to that time the next day.

LINE <line width> may be specified as an integer in the range 72 through 255, to set the maximum terminal width (the MAXRECSIZE of the session output file). The default is the input file text length or 72 characters, whichever is larger.

The scheduled session runs with the usercode that scheduled the session. The chargecode may be explicitly set or, by default, the chargecode is the chargecode in effect at the time the SCHEDULE command was processed. The user's default FAMILY (obtained from the USERDATAFILE) is used to begin the scheduled session; the FAMILY in effect when the SCHEDULE command was processed is not used. The FAMILY command may be used to change the FAMILY during the scheduled session.

SCHEDULE Input

By default, each input line is considered to end after the last nonblank character. If the EOL <character> is specified, that character may be used to mark the end of an input line by placing it after the last column of valid input on the line. All further information is ignored because the EOL character is treated as a "stopper" character for the scanner. The <character> may be any nonblank, nonalphanumeric graphic, or the ASCII mnemonic for a control character. For example, using the specification EOL permits the following input lines:

```
FIX 1230/e./e.
1460
```

Using the above EOL specification also allows a FIX insertion to end in a blank and allows input of a blank line, respectively. On terminals that can input control characters, usurpation of any graphic character can be avoided if the control character chosen does not interfere with the Data Comm usage of that station. SUB is a suitable control character that is entered as control-Z on most ASCII keyboards and may be entered via EBCDIC cards as a 7-8-9 multipunch.

In essence, the schedule input file is a script for a session. The schedule input file must contain every input line that the user would type at the terminal (after logging-on) if the session were interactive. The schedule input file may contain any CANDE command except HELLO, PASSWORD, or TAPE. Any TERM command must be empty (containing no change to terminal specifications). A BYE command is optional and terminates the session wherever it appears. CANDE infers a ?END and/or BYE, if appropriate, at the end of the schedule input file. A line beginning with a question mark (?) is treated as a control command; the control commands acceptable in a schedule session are defined in subsequent text in this section. A line entered in automatic-sequencing mode and beginning with two question marks (??) is treated as valid data beginning with a question mark (?), as in interactive CANDE sessions. Also, a line beginning with two question marks (??) being read by a program has the first question mark (?) discarded; the line is then given as data to the program. This feature is not available to an interactive session.

REMOTE Files

A schedule session is run with a dummy Data Comm station to provide for REMOTE files. For most purposes, the schedule station behaves programmatically like a real station. Specifically, the REMOTE files of any tasks processed from the schedule session are, by default, associated with the schedule station. As with any session, this linkage is accomplished through the STATION task attribute and may be overridden by explicit user action. By setting STATION to a valid Logical Station Number (LSN), or by setting it to zero and equating the file TITLE appropriately, a task of a schedule session may attempt to open a file on a genuine Data Comm station (subject to the same constraints as any other "foreign" user of a station). On the other hand, only tasks of the particular schedule session may open REMOTE files to a schedule station.

Schedule Control Commands

The following control commands are available for schedule as well as interactive sessions:

?END Denotes the end of input for a program.
 ?TIME Show current time and date.
 ?% Enters comments.

The following special control commands are available only for schedule sessions:

?REPORT Sends message to interactive user.
 ?NUMBERED Lists input-file sequence numbers in output file.
 ?UNNUMBERED Does not list input-file sequence numbers in output file.
 ?RESTART Restarts aborted sessions.
 ?NORESTART Does not restart aborted sessions.
 ?RESUME Resumes input after schedule session errors are detected.
 ?EOL Changes end-of-line character.

For a description of these commands refer to Section 5, CONTROL COMMANDS.

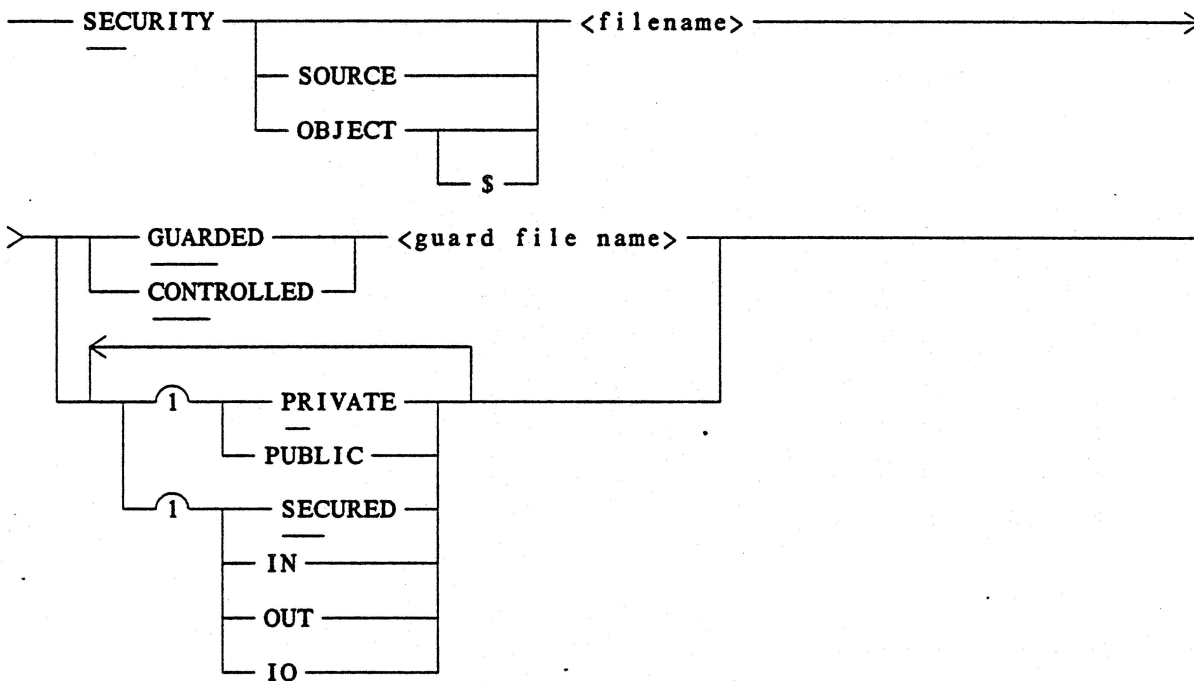
Each schedule session is assigned a unique integer for identification and operator interaction called "schednum". Schednum is always displayed as a five-digit number (with leading zeros) to distinguish it from a session number. The schednum is assigned when the schedule input file is created and is included in the schedule input file title. The schednum appears in the output file and in messages about the schedule session. A session number is assigned when the schedule session processing begins; if the session is restarted or SPLIT, a new session number is assigned. All session numbers used by a schedule session appear in the second line of the output file, as well as in the separate log-on or session-split messages.

Examples

```
SCH SCH/WFL
#SCHEDULE OUTPUT FILE NAME IS IN USE
#FILE:(UZER)SCHOUT/SCH/WFL ON USERPACK
#
REMOVE SCHOUT/SCH/WFL
#(UZER)SCHOUT/SCH/WFL ON USERPACK REMOVED
SCH SCH/WFL
#SCHEDULE #00009 RUNNING
  OUTPUT FILE IS (UZER)SCHOUT/SCH/WFL ON USERPACK
  LINE WIDTH IS 72
#
```

SECURITY

Syntax



Semantics

The SECURITY command allows the security attributes of a file in a library to be specified.

<file title> specifies the file whose security attributes are to be changed. If neither the SOURCE nor OBJECT options are specified, the security attributes are changed for the <file title> and OBJECT/<file title> of the file. The SOURCE option changes attributes for <file title> only and does not change the OBJECT/<file title>. The OBJECT option changes only the attributes of OBJECT/<file title> and does not change the source <file title>. An object file not under the object directory as OBJECT/<file title> can be specified by using the OBJECT \$ form, or the simple form:

```
SECURITY <file title> ...
```

The security options that can be specified are identical to the file attributes SECURITYUSE and SECURITYTYPE. The SECURITYTYPE options are PRIVATE and PUBLIC (or CLASSA), and the SECURITYUSE options are SECURED, IN, GUARDED, CONTROLLED, OUT, and IO. Brief explanations of these options are given in the following paragraphs.

CANDE REFERENCE MANUAL

PRIVATE

The PRIVATE option specifies that the file may be accessed only by the owner or a privileged user.

PUBLIC

The PUBLIC option allows access to the file to any user who references the file using the <usercode><file name> form of the file title.

GUARDED

The GUARDED option allows the security of files with the file name specified to be changed to GUARDED.

CONTROLLED

The CONTROLLED and GUARDED options are synonyms. Changing the security from CONTROLLED to any other security requires the file to have an access of READ/WRITE.

SECURED

The SECURED option implies that the file may not be accessed; the only exception is that the file may be executed if it is an object code file.

IN

The IN option indicates that read-only access is allowed.

OUT

The OUT option indicates that write-only access is allowed.

IO

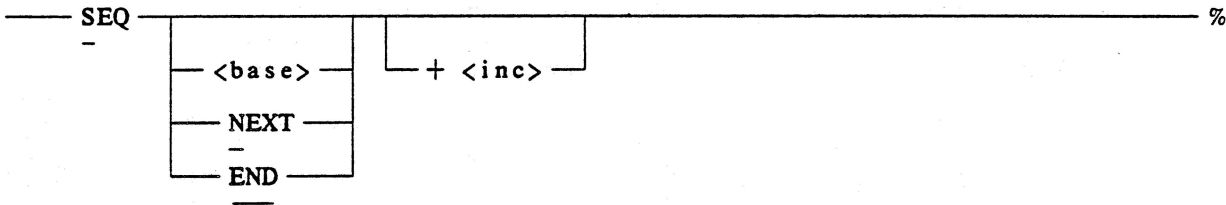
The IO option indicates that read or write access is allowed.

Example

```
SECURITY CANDE/MANUAL PUBLIC IO
# (UZER)CANDE/MANUAL ON USERPACK SECURITY CHANGED
```

SEQ

Syntax



Semantics

The SEQ command causes the system to automatically provide a sequence number for each line entered in the workfile. While in the automatic sequencing mode, the system recognizes only the commands that begin with the control character (control commands) of the station and a special adaptation of the MARGIN and FIX commands; all other information is treated as text. Automatic sequencing continues until the system receives a line without text or margin specifications. (Refer to the MARGIN and FIX commands in this section for a description of their use in sequence mode.)

The starting line number can be specified in one of three ways, as defined in the following paragraphs:

<base>

<base> explicitly defines the starting line number.

NEXT

NEXT sets the starting line number to the next line number that would have resulted from the last SEQ, RESEQ, MOVE, or INSERT command. If none of these commands have been issued, then 100 is used.

END

END sets the starting line number to the largest line number in the workfile plus the specified (or default) increment. If no specifications appear, 100 is used.

<inc>

<inc> specifies an increment to be used in generating subsequent line numbers. 100 is the default unless an increment has been specified in a prior SEQ, RESEQ, MOVE, or INSERT command.

CANDE REFERENCE MANUAL

Pragmatics

In automatic sequence mode on a teletype or similar device, the new line number is typed at the beginning of each line; the text desired for that line is then entered. On devices with multiline input capability, a new sequence number is displayed following each transmission; the line number is incremented but not displayed for subsequent lines within a transmission block.

Sequence mode is terminated by entering a null line. CANDE responds with a pound sign (#) to note the return to normal mode. For multiline transmission, the lines must be separated with carriage returns.

Normal CANDE commands are not recognized in sequence mode because they are indistinguishable from text entries. If the line begins with the control character (?), the input is processed as a control command. If a "?" is to be entered as the first character of a line in sequence mode, the line must be preceded by an extra question mark (?). That is, a question mark (?) must appear twice (??) if it is to be entered as the first character.

Although sequence numbers are generated, the line is otherwise treated as a single-line entry. That is, if the new line number matches one in the workfile, the newer line replaces the older.

Different types of terminals may act differently in automatic sequence mode; therefore, the characteristics of the particular terminal to be used should be learned.

Examples

```
S
100YOUR STATEMENTS
200@5
300TEST COLUMNS
400
#

L
100 YOUR STATEMENTS
200     TEST COLUMN
#

seq 10+3
10 your statements....
13 and so on....
16 and so on....
19
#
```


CANDE REFERENCE MANUAL

Margin Specifications - the () option

The column at which the text is to begin can be specified by use of the (@) option. If this option is not specified, the default is column 1. An at sign (@) followed by only a colon indicates that the text is to begin in column 1. Thus, if the current margin setting is 5, it allows the text of the one line to begin at column 1. The (@) option followed by an <integer> specifies that the text is to start in the column specified by <integer>. Thus, the following command enters a line in the workfile at sequence number 100 with the word "text" beginning in column 5:

```
100@5:text
```

The (@) option followed by +<integer> or -<integer> specifies that the text is to begin plus or minus the number of columns specified by <integer>, relative to the current margin setting. For example, the following command enters a line in the workfile at sequence number 100 with the word "text" beginning five columns to the right of the current margin setting:

```
100@+5:text
```

If the current margin setting is 10, then "text" appears starting in column 15.

NOTE

To permanently set the margin to a particular column, the MARGIN command must be used.

Examples

```
MAKE TEST1 DATA
#WORKFILE TEST1:SEQ
25begin
100this is a single-line entry
215and so is this
25begin
300@5:this text starts in column 5
315@+10:this text starts in col. 11 (col. 1 + 10 = 11)
317@@ this text begins with a single "@" character
000003208 digits maximum in type seqdata files
```

CANDE REFERENCE MANUAL

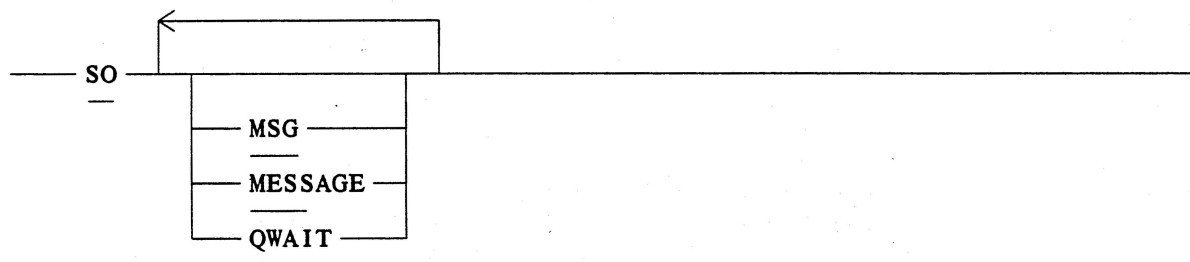
```
list
#WORKFILE TEST1
25 begin
100 this is a single-line entry
215 and so is this
300     this text starts in column 5
315     this text starts in column 11 (col 1 + 10 = 11)
317 @ this text begins with a single "@" character
320 8 digits maximum in type seqdata files
#
```

```
215
300line 215 was just deleted, and i replace the line at 300
```

```
list
#WORKFILE TEST1
25 begin
100 this is a single-line entry
300 line 215 was just deleted, and i replace the line at 300
315     this text starts in column 11 (col 1 + 10 = 11)
317 @ this text begins with a single "@" character
320 8 digits maximum in type seqdata files
#
```

SO

Syntax



Semantics

The SO and RO commands control the setting and resetting of station options. SO sets station options, and RO resets station options. The current state of station options is displayed if either the RO or SO command is entered without a following station option name.

MESSAGE and MSG are synonyms. These options control the automatic display of job- and task-related messages at the terminal during a CANDE session. If MSG is set, job- and task-related messages are automatically displayed. When MSG is reset, only RSVP and DS messages are displayed.

The QWAIT option specifies that the state of the user's queued input is to be set to waiting rather than pending in the event of an error. (Refer to Queue Manipulation in Section 2 for details.)

Examples

```

SO
#MESSAGES SET, QWAIT RESET
so messages
#MESSAGES SET
SO QWAIT
#QWAIT SET
  
```

SPLIT

Syntax

SPLIT

Semantics

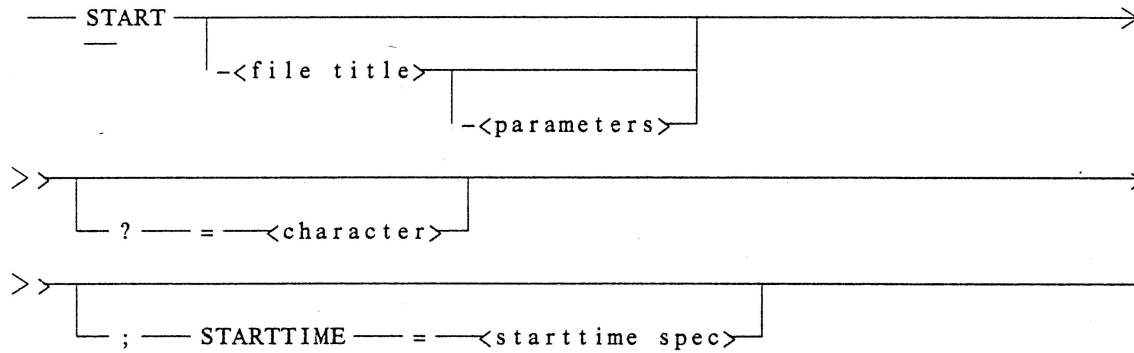
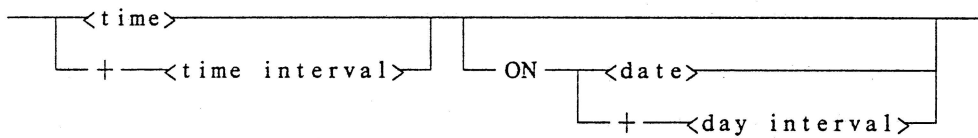
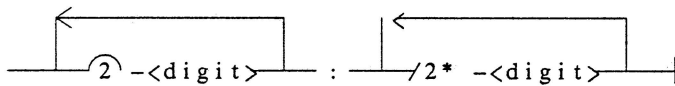
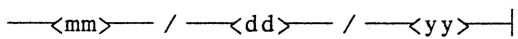
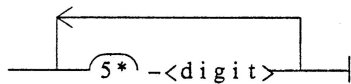
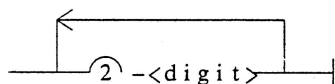
The SPLIT command terminates the current session and automatically begins another session under the same usercode. The workfile, if any, is not disturbed. If the current session has produced any printer or card punch output, that output is queued for printing/punching at this time.

Example

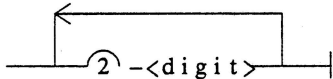
```
SPLIT
#SPLIT SESSION 7562 ET=1:23:57.3 PT=13.6 IO=6.8
#NEW SESSION 7695 11:08:20 05/08/78
```

START

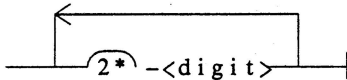
Syntax

`<starttime spec>``<time>``<date>``<julian date>``<m>`

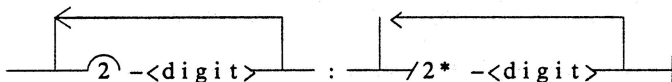
<d>



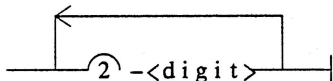
<y>



<time interval>



<day interval>



Semantics

The START command invokes the WFL compiler to process the specified <file title>. <parameters> is any arbitrary list of constants (that is, integer, real, string, or Boolean) recognized by WFL. If <file title> is not specified, the current workfile is assumed. By default, the question mark (?) is interpreted by WFL to be an invalid character when found in column one of an input line. This default may be changed by specifying ?=<character>, where <character> represents the character that WFL is to interpret as the invalid character.

The resulting job is invoked separately from the session, although progress may be monitored through use of ?JA, ?MIX, ?C, ?MSG, and so forth.

The invoked WFL compiler determines the absolute time and date at which the job should begin execution from the <starttime spec>. This start time overrides any <starttime specification> that is made in the <job attribute specification> list of the job that is being started. (Refer to JOB ATTRIBUTE SPECIFICATION in Section 5 of the B 5000/B 6000/B 7000 Series Work Flow Language Reference Manual.)

<time> is the time of day on a 24 hour clock in the form HH:MM. HH, the hour, must be less than 24, and MM, the minute, must be less than 60.

<date> is the date in the form MM/DD/YY.

<julian date> is the date in the form YYDDD, where YY is the year and DDD is the number of the day of the year.

<time interval> is of the form HH:MM. HH, the number of hours, must be less than 24, and MM, the number of minutes, must be less than 60.

If a <day interval> is specified, that number of days is added to the current date.

Examples

```
START WFL/1
#RUNNING 7768
#JOB 7769 IN Q 04
#
#7769 BOJ TESTJOB
#7769 DISPLAY:IN START JOB
#7769 7769 EOJ JOB TESTJOB
L TEST
#FILE (UZER)TEST ON USERPACK
100 ?BEGIN JOB TEST/PARAM (INTEGER I, STRING S);
200 INTEGER J;
300 DO
400 BEGIN DISPLAY S; J:=J+1; END
500 UNTIL J=I;
600 ?END JOB
#
START TEST(2,"HELLO")
#RUNNING 4833
#JOB 4834 IN Q 04
#
#4834 BOJ TEST/PARAM
#4834 DISPLAY:HELLO.
#4834 DISPLAY:HELLO.
#4384
#4384 4384 EOJ TEST/PARAM
START WFL/EX/1; STARTTIME= 23:00
#RUNNING 1101
#JOB 1102 IN QUEUE 0
#
START WFL/EX/2; STARTTIME= + 0:05
#RUNNING 1103
#JOB 1104 IN QUEUE 0
START WFL/EX/3; STARTTIME = 1:30 ON 08/29/81
#RUNNING 105
#JOB 1106 IN QUEUE 0
#
START WFL/EX/4; STARTTIME = 1:00 ON + 2
#RUNNING 1107
#JOB 1108 IN QUEUE 0
#
```

STATUS

Syntax

```
— STATUS —<output filename>—|
```

Semantics

The STATUS command may be used to inquire about any schedule session initiated by the user. If the session is not yet in progress, #SCHEDULE #<schednum> IS SCHEDULED is displayed; if an AFTER <time> is pending, that information is also displayed.

If the session is in progress, information similar to the following is returned:

```
#SCHEDULE #00017 STARTED AT 12:34
  READ 12 (THRU 2100), WRITTEN 146, RESUMED AFTER 2 ERRORS
  SESSION 5214, TASK/<lsn> 5215/119
```

The schednum, start time, lines read and written, sequence number last read, and session number always appear. The error count appears if it is nonzero.

If the session has been completed, a message indicates normal termination; if the session has been abnormally terminated, a message indicates the nature of abnormal termination.

A STATUS inquiry made during the first minute after CANDE is restarted may show a status like #SCHEDULE #00008 STARTED AT 09:27. No further information is displayed because the session was running when CANDE was aborted and the schedule facility has not yet been reinitialized. As soon as schedule initialization is completed, the status is changed to AWAITING RESTART or ABORTED BY SYSTEM FAILURE; NO RESTART.

NOTE

The minimum abbreviation is STAT. The STATUS command is distinct from the ?STATUS control command. Inadvertent abbreviation of STAT produces a START command.

Example

```
STATUS SCHOUT/SCH/10
#SCHEDULE #00014 STARTED AT 13:07
  READ 5 (THRU 500), WRITTEN 19
  SESSION 7958, TASK/<lsn> 7960/39
```

STOP**Syntax**

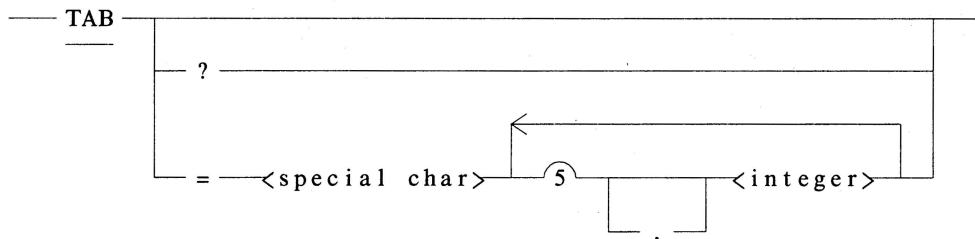
STOP <output filename>

Semantics

The STOP command causes the schedule session associated with <output filename> to be aborted or removed from the queue if it is in progress or pending, respectively.

Example

```
STOP SCHOUT/SCH/10
#SCHEDULE #00014 BEING STOPPED
```

TAB**Syntax****Semantics**

The TAB command allows the user to define a tab character and set tab columns. These tabs have no relation to the TAB functions on a device. <special char> may be any special character except (?), (@), (*), and <>null>.

The <integer> values specify the columns in which the cursor is to be placed (that is, the same numbers in as the MARGIN command). When entering a source line, the tab characters are inserted whenever the user wishes to JUMP to a tab column. If a jump to the first tab column is desired, a single tab character is inserted in the text. If the second tab column is desired, two tab characters are inserted unless a tab character had been inserted previously to jump to the first tab column. In the latter case, only one additional tab character is inserted. If the number of tab characters seen indicates a backwards skip, a warning message is displayed and the tab is ignored. If CANDE is in sequence mode, sequence mode is terminated (as in a line overflow).

A TAB command with no parameters clears the current tab settings. The TAB? command displays the current tab settings. If the character is not a graphic character, its name or HEX representation is displayed (for example, HT, VT, CB).

Tabs are cleared at each new log-on or via the TAB command entered by itself.

CANDE REFERENCE MANUAL

Examples

The pound sign (#) indicates a blank character; input is to a workfile.

COMMAND: TAB = 5,10,15 [TABS CHAR= , TABS AT COL 5,10,15] 1. INPUT: ABC
ACTION: ####ABC

2. INPUT: ABC DEF
ACTION: ####ABC##DEF

3. INPUT: ABCD EFG
ACTION: ABCDEFG

4. INPUT: ABCDE FG
ACTION: ABCDEFG
WARNING: #TAB COL(5) < CURRENT COL(6) - IGNORED.

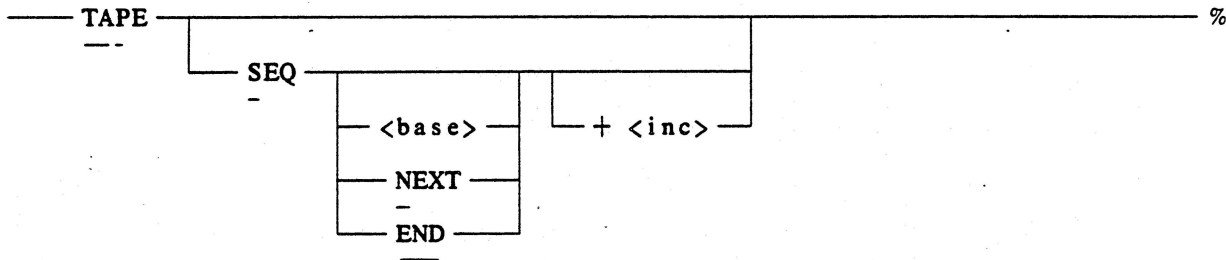
5. INPUT: ABCDE FG
ACTION: ABCDE####FG

COMMAND: TAB?
RESPONSE: #TAB = 5,10,15

COMMAND: TAB
RESPONSE: #TABS CLEARED

TAPE

Syntax



Semantics

The TAPE command initiates reading data from paper tape. The data are entered in the workfile as it is read.

After the tape command is entered, CANDE sends #OK followed by a carriage return (CR), line feed, and DC1 (XON). The XON automatically starts the paper tape reader on some terminals; on other terminals, the reader must be manually started within 15 seconds. If 15 seconds elapse with no input received, the Data Communications Processor signals an error termination, and CANDE sends a message to the terminal. Once input has begun, it continues until the paper tape reader stops (as noted by a 500-millisecond period with no new character received). During input, a CR (carriage return) delimits a workfile line. The following character codes are ignored or discarded:

- LF (line feed)
- NUL
- NUL (rubout)
- DC1 (XON)
- DC3 (XOFF)

Sequence numbers are optional on the input tape; however, if the tape does not contain sequence numbers, the SEQ option must be used. The SEQ option causes CANDE to supply sequence numbers for all lines. (Refer to the SEQ command in this section for semantics regarding the syntactic items of this option.)

Because the text field begins at the first nondigit or the first character after the maximum number of digits for a sequence number, leading zeros in sequence numbers must be supplied only if the first character of text is a digit.

CANDE REFERENCE MANUAL

Input Errors

Paper tape input is terminated if no input appears for 15 seconds. The input following the error is discarded (1) if a break is detected on the line; (2) if a record exceeds maximum input length; or (3) if the Data Communications Processor fails to acquire message space when needed. A stop bit error is treated by substituting a question mark (?) for the garbled character; input is not aborted.

Off-line Preparation of Paper Tape Input

The minimum requirement for CANDE input is that each line be terminated by a CR. In order for the output to be readable on the teletypewriter as the tape is being read, each line should be terminated by CR, LF, DEL rubout, or NUL. The LF advances the paper, and the rubout or NUL gives the carriage time to return. Leader or trailer tape can be punched using the repeat key with DEL or NUL. (NUL can be punched on many teletypewriters by using shift, control, and P.) Errors committed in punching the tape can be corrected by backspacing the paper tape, typing rubout, and then typing the correct character. (Usually a backspace exists on the paper tape punch for this purpose.)

To have the system punch a paper tape of an existing file, use the PUNCH option of the LIST command.

Examples

```
M DON/1
#WORKFILE DON/1: SEQ
TAPE SEQ
#OK
LIST
100 THIS WAS AN EXAMPLE.
200 END.
#
```

```
MAKE T; TAPE
#WORKFILE T: SEQ
#OK
#WORKFILE T
00000100THIS IS THE FILE BEING READ IN
00000200IT WAS LISTED WITH THE L : PUNCH COMMAND
```

```
#1 RECORDS DISCARDED (NO SEQUENCE NUMBER)
#
L
100 THIS IS THE FILE BEING READ IN
200 IT WAS LISTED WITH THE L : PUNCH COMMAND
#
```


CANDE REFERENCE MANUAL

CONTINUOUS causes CANDE to send continuously until end-of-output.

When an output line exceeds the terminal width, CANDE either truncates the line or "folds" it by printing it on several lines. A special foldmark character is printed as the last character of a truncated or interrupted line, and as the first character of a continuation line. The foldmark character may be specified by placing it in quotes in the TERMINAL specification command. If the quotes enclose more than one character, the first is used.

If TERMINAL is entered with no options specified, CANDE displays the current terminal settings.

Examples

```
TERMINAL
```

```
#LINE = 80,PAGE = 24,BUF = 1920,SCREEN,WAIT," "
```

```
term line 72
```

```
#LINE = 72,PAGE = 24,BUF = 1920,SCREEN,WAIT," "
```


CANDE REFERENCE MANUAL

TITLE <directory name>/= TO <new directoryname>/=

The TITLE <directory name>/= TO <new directoryname>/= variation changes the name of the <directory name> to <new directoryname> and also changes OBJECT/<directory name>, if it exists, to OBJECT/<new directoryname>.

TITLE SOURCE <file name> TO <new filename>

The TITLE SOURCE <file name> TO <new filename> variation changes only the source file <file name> to <new filename>. Any object file associated with <file name> is unchanged.

TITLE SOURCE <directory name>/= TO <new directoryname>/=

The TITLE SOURCE <directory name>/= TO <new directoryname>/= variation changes the <directory name> to <new directoryname>. Any associated directory under the object directory is unchanged.

TITLE OBJECT <file name> TO <new filename>

The TITLE OBJECT <file name> TO <new filename> variation changes the name of OBJECT/<file name> to OBJECT/<new filename>. The dollar sign (\$) option can precede either <file name> or <new filename>, or both. For <file name>, the dollar sign (\$) indicates to CANDE that the file is not stored as OBJECT/<file name> but is stored as <file name>. When a dollar sign (\$) precedes <new filename>, CANDE changes the file name to <new filename>, rather than OBJECT/<new filename>.

TITLE OBJECT <directory name>/= TO <new directoryname>/=

The TITLE OBJECT <directory name>/= TO <new directoryname>/= variation changes the name of OBJECT/<directory name> to OBJECT/<new directoryname>. The dollar sign (\$) option can precede either <directory name> or <new directoryname>, or both. For <directory name>, the dollar sign (\$) indicates that the directory is not stored as OBJECT/<directory name> but as <directory name>. When a dollar sign (\$) precedes <new directoryname>, CANDE changes the name of the directory to <new directoryname>, rather than OBJECT/<new directoryname>.

CHANGE and TITLE are synonyms.

Examples

The following example renames the source file A to B and the object file OBJECT/A to OBJECT/B.

```
TITLE A TO B
# (UZER)A CHANGED TO (UZER)B ON USERPACK
```

The following example renames the directory PROGRAMLIBRARY to LIBRARY. Also, the directory OBJECT/PROGRAMLIBRARY is renamed to OBJECT/LIBRARY. Thus, a source program that was once under the PROGRAMLIBRARY directory (PROGRAMLIBRARY/MAINT, for example) is now referenced as LIBRARY/MAINT. The object code file for PROGRAMLIBRARY/MAINT is now stored as OBJECT/LIBRARY/MAINT.

```
CHANGE PROGRAMLIBRARY/= TO LIBRARY/=
# (UZER)PROGRAMLIBRARY/= CHANGED TO (UZER)LIBRARY/= ON USERPACK
```

CANDE REFERENCE MANUAL

The following example renames the user s object file TEST (stored as OBJECT/TEST) to TEST (stored as TEST).

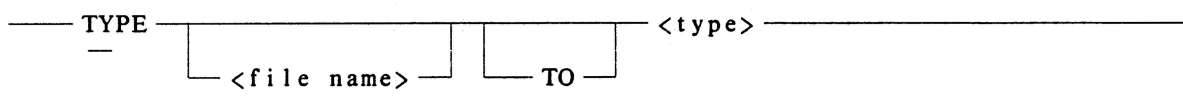
```
TI OBJECT TEST TO $ TEST
```

The following example renames the object file TESTONLY (which is not stored under the OBJECT directory) to OBJECT/TESTONLY.

```
TI OBJECT $ TESTONLY to TESTONLY
```

TYPE

Syntax



Semantics

The TYPE command changes the FILEKIND attribute of the specified file. If <file name> is not specified, the workfile is assumed. <type> may be any of the names specified in Section 3, BASIC CONSTRUCTS, under the heading <type>.

If the <file name> is used, it must be in the library of the user (or usercode); and, if the type of the existing file is incompatible with the new type (different record size, sequence number, and so forth), the request is rejected.

Examples

```

type algol
#
TYPE DATAFILE COBOL
#
ty fortran/file seq
#

```

UPDATE

Syntax

UPDATE

Semantics

The UPDATE command forces the system to update the workfile immediately. All changes, additions, and deletions accumulated are applied to the workfile, and an updated workfile is generated. CANDE acknowledges the request by displaying #UPDATING and then signifies completion of the update by displaying a pound sign (#).

The UPDATE command is invoked implicitly whenever needed to permit another command to function properly; therefore, the UPDATE command is not normally entered on the terminal explicitly. The UPDATE command is implicitly invoked when any of the following commands are encountered: BIND, COMPILE, EXCLUDE, EXECUTE, INSERT, MERGE, MOVE, RANGE, REPLACE, RESEQ, RMERGE, RUN, SAVE, START, TYPE, WRITE, and UTILITY.

However, at least three cases exist in which the UPDATE may require explicit invocation. These three cases are described in the following text.

1. The UPDATE may be explicitly invoked in order to assure that a workfile is a private copy. If user X gets a file from a library other than the library for his usercode (for example, GET (Y)HISFILE), the workfile source resides in the library of user Y and is subject to any change or removal done by Y on that file until an UPDATE is invoked (either explicitly or implicitly) by user X. The update action creates a separate copy of the file for user X in a workfile area.
2. The UPDATE may be explicitly invoked in order to assure a complete workfile recovery in the event of a system failure. If the system fails during during the modification of a workfile, the last several changes to the workfile (a maximum of four) may be lost. This loss occurs because CANDE defers changes to a workfile as long as possible before invoking the time-consuming UPDATE. If the workfile is updated when a system failure occurs, the workfile is recovered completely.

CANDE REFERENCE MANUAL

3. The UPDATE may be explicitly invoked in order to assure that a complete update of the file is accomplished. That is, the implementation of CANDE is such that if a sufficient number of changes (none requiring an update) is made to the workfile, more than one update may be required to accomplish a complete update of the file. This UPDATE use produces a file that represents all outstanding workfile changes. In the case where the workfile is either type DATA or CDATE, this process sometimes results in an incorrect update. CANDE discards any outstanding workfile changes that cannot be made with one update of the file and displays the following error message:

DATA LOST, LIMITED CHANGES ALLOWED TO DATA FILES

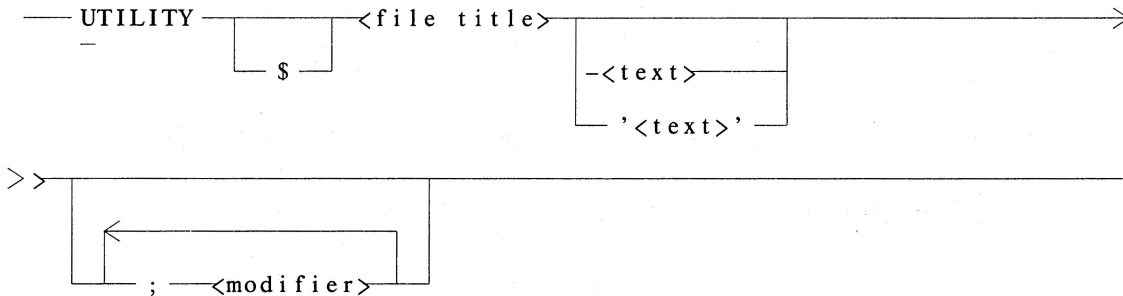
To prevent an incorrect update from occurring, a periodic update of a DATA or CDATE file must be performed using the UPDATE verb.

Example

```
UPDATE
#UPDATING
#
```

UTILITY

Syntax



Semantics

The `UTILITY` command allows user-written programs to modify the current workfile. The utility to be used is specified by name (as in executing any other program) and may optionally provide a string of text that is passed as an array parameter to the utility. CANDE makes the workfile available to the utility for access or modification using file equation and the task value for communication. The `<text>` is any string of data not including a semicolon. One or more `<modifier>`s may be included (as for the `EXECUTE` verb).

The dollar sign (`$`), `<file title>`, and `<modifier>`s are treated the same as for the `EXECUTE` verb. The `<text>` is copied into an array that is passed as a parameter to the utility. Except between pairs of quotes, any lower-case letter is translated to upper case and any redundant blanks are discarded. (A blank is considered redundant if it is both preceded and followed by a character that is not a letter or digit, or if it ends the `<text>`.) If any alterations are made to the workfile, the workfile is updated before the utility is invoked.

The utility must be a procedure accepting an array parameter, as in the following ALGOL heading:

```

$ SET LEVEL 2
PROCEDURE U(A); ARRAY A[*];

```

The array contains the text from the utility command, in EBCDIC, followed by at least one NUL. The length of the array is arbitrarily large; all characters beyond text are NUL.

CANDE REFERENCE MANUAL

At most, five files may be label-equated. The following list illustrates the five files that may be label-equated.

Intname -----	Attribute -----	Specification -----
WORKFILE	TITLE	Name of workfile
WORKSOURCE	TITLE	Saved or unsaved worksource
NEWWORKSOURCE	TITLE INTMODE EXTMODE UNITS MAXRECSIZE BLOCKSIZE FILEKIND SECURITYTYPE SECURITYUSE SECURITYGUARD CYCLE VERSION SAVEFACTOR	Potential new unsaved worksource
WORKOBJECT	TITLE	Saved or unsaved workobject
NEWWORKOBJECT	TITLE	Potential new unsaved workobject

If the CANDE session has no workfile, no label equation is done. If a workfile exists, it may or may not be named. If a workfile exists and is named, then WORKFILE.TITLE is equated to the workfile name. The name has neither usercode prefix nor family suffix and serves only for identification. The utility is not expected to perform any input or output via the file WORKFILE.

A workfile may be empty. If the workfile is not empty, a worksource file is made containing the line images for the file. If a worksource exists, then WORKSOURCE.TITLE is equated to its title. A utility reads this file to obtain the workfile contents. To be consistent with CANDE conventions, the utility should not write to WORKSOURCE.

If the workfile has been compiled, the object code is contained in a workobject file. If a workobject exists, WORKOBJECT.TITLE is equated to its title. A utility could read this file to display or analyze the code.

If any workfile (named or not, empty or not) exists, label equation is performed for the files NEWWORKSOURCE and NEWWORKOBJECT. All the attributes listed above for NEWWORKSOURCE are equated to the appropriate values for the workfile. The utility may write a new worksource with these attributes to replace the old worksource file. The TITLE of NEWWORKOBJECT is equated; a file of this name may be created if the utility invokes (or is) a compiler to create a new workobject.

CANDE REFERENCE MANUAL

All the source and object titles are complete, including usercode prefix and family suffix, except that ON DISK cannot appear for files on disk.

On entry to the utility, the utility TASKVALUE is set as follows:

```
Bits 47:44 = 0.
Bit 0:1 = 1, if a workfile exists.
Bit 1:1 = 1, if a workfile name exists.
Bit 2:1 = 1, if a worksource exists.
Bit 3:1 = 1, if a workobject exists.
```

On termination of the utility, CANDE interprets the TASKVALUE as follows:

```
Bit 46:1 = 1, an error occurred; CANDE terminates or RESUMES a
              SCHEDULE session, or holds pending queued input.
              The utility must generate its own error message as
              CANDE does not. Bits 23 and 22 remain significant
              when bit 46 is set.
Bit 22:1 = 1, a new worksource has been written, using the title
              and other attributes supplied for NEWWORKSOURCE.
Bit 23:1 = 1, a new workobject has been written, using the title
              supplied for NEWWORKOBJECT.
```

If the utility is abnormally terminated, CANDE ignores the TASKVALUE and acts as though bit 46 were on and all others off.

Examples

In the following example, a CANDE user with usercode = MY and family specification of DISK = MINE OTHERWISE DISK, using station number 12, with access to a utility program named OBJECT/P is assumed. The following session fragments result in the utility being run with the attributes noted. Those TITLE attributes not mentioned in each case default to the INTNAME. In each case that NEWWORKSOURCE.TITLE is specified, all the other attributes listed previously have been equated to the values appropriate to the workfile.

No workfile exists.

```
REM;U P          MYSELF.TASKVALUE=0
```

A named, empty workfile exists.

```
MAKE T;U P      MYSELF.TASKVALUE=3
                WORKFILE.TITLE="T."
                NEWWORKSOURCE.TITLE="(MY)CANDE/TEXT120 ON MINE."
                NEWWORKOBJECT.TITLE="(MY)CANDE/CODE120 ON MINE."
```

A named workfile with an unsaved worksource and workobject exists.

```
MAKE T          MYSELF.TASKVALUE=15
100BEGIN        WORKFILE.TITLE="T."
...            WORKSOURCE.TITLE= "(MY)CANDE/TEXT120 ON MINE."
900END.         WORKOBJECT.TITLE= "(MY)CANDE/CODE120 ON MINE."
C              NEWWORKSOURCE.TITLE="(MY)CANDE/TEXT120 ON MINE."
U P            NEWWORKOBJECT.TITLE="(MY)CANDE/CODE120 ON MINE."
```

CANDE REFERENCE MANUAL

- A named workfile with a saved worksource and workobject exists.
GET Z;U P
MYSELF.TASKVALUE=15
WORKFILE.TITLE="Z."
WORKSOURCE.TITLE="(MY)Z ON MINE."
WORKOBJECT.TITLE="(MY)OBJECT/Z ON MINE."
NEWWORKSOURCE.TITLE="(MY)CANDE/TEXT120 ON MINE."
NEWWORKOBJECT.TITLE="(MY)CANDE/CODE120 ON MINE."
- A named workfile with a "borrowed" worksource exists.
G (U)X ON F AS W
U P
MYSELF.TASKVALUE=7
WORKFILE.TITLE="W."
WORKSOURCE.TITLE="(U)X ON F."
NEWWORKSOURCE.TITLE="(MY)CANDE/TEXT120 ON MINE."
NEWWORKOBJECT.TITLE="(MY)CANDE/CODE120 ON MINE."
- An unnamed workfile with a "borrowed" worksource exists.
G SYMBOL/ALGOL
U P
MYSELF.TASKVALUE=5
WORKSOURCE.TITLE="*SYMBOL/ALGOL."
NEWWORKSOURCE.TITLE="(MY)CANDE/TEXT120 ON MINE."
NEWWORKOBJECT.TITLE="(MY)CANDE/CODE120 ON MINE."

WFL

Syntax

WFL — <work flow language statement(s)> —————

Semantics

The WFL command invokes the Work Flow Language (WFL) compiler to process the <work flow language statement(s)>. If the specified <work flow language statement(s)> are free of syntax errors, the resulting job is invoked.

Examples

```
WFL DISPLAY "STRING";  
#RUNNING 8114  
#BOT 8115 (UZER)WFLCODE  
#8115 DISPLAY: STRING.  
#EOT 8115 (UZER)WFLCODE  
#
```

WHAT

Syntax

— WHAT —|

Semantics

The WHAT command indicates the state of the workfile. The amount of information provided varies depending on the current state of the workfile; the amount of information relevant or available is indicated. The output may include the following information:

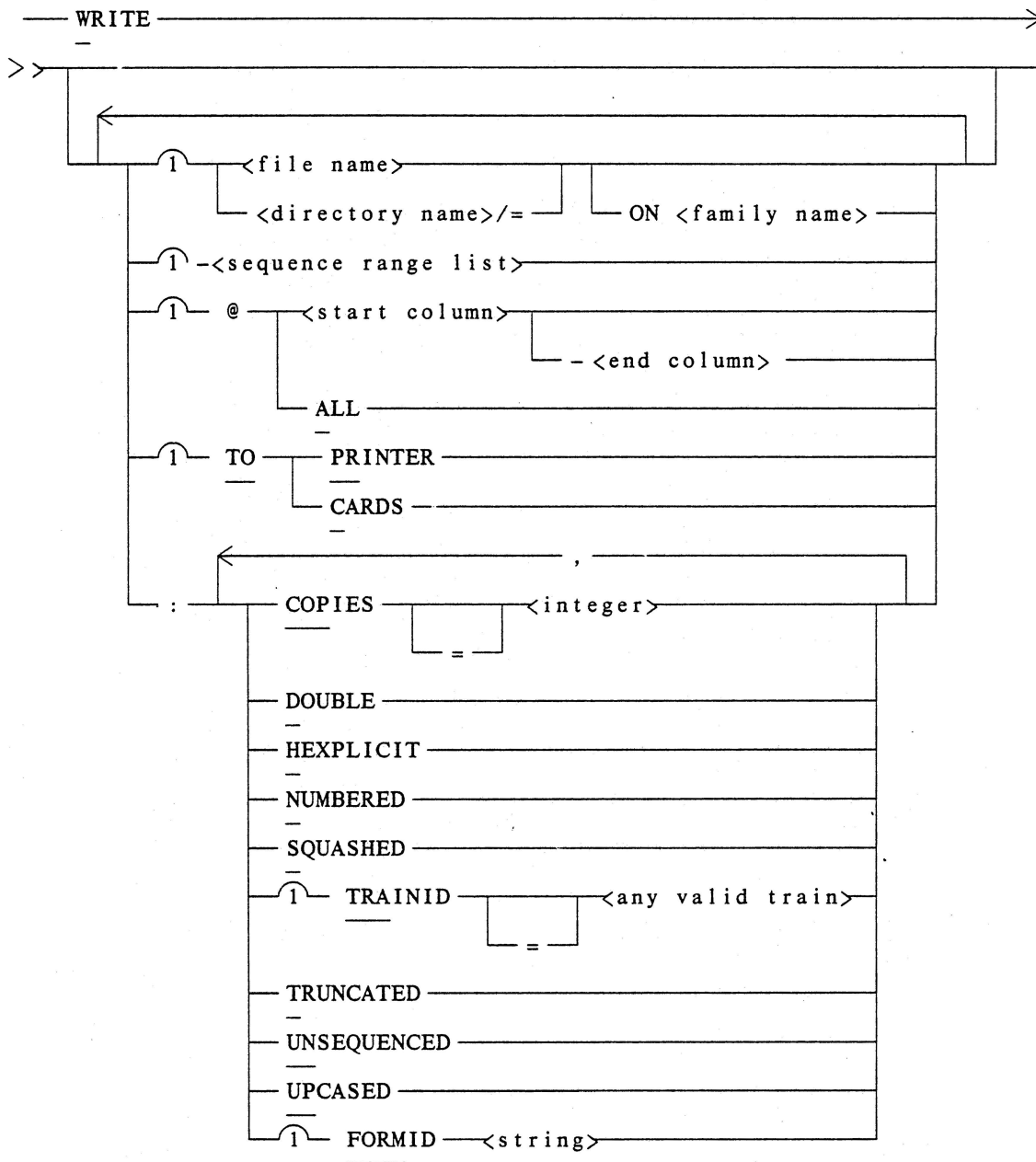
1. Title (name)
2. Type (filekind)
3. Number of records
4. Sequence number of last record
5. Object file presence
6. Saved or unsaved status

Example

```
WHAT  
#WORKFILE PAYROLL/INFO: SEQ, SAVED
```

WRITE

Syntax



CANDE REFERENCE MANUAL

Semantics

The WRITE command allows a file (or portions thereof) to be listed on the line printer or punched on cards. As with any output generated by a user, backup files created by using the WRITE verb are not actually printed until the user either logs-off or uses the SPLIT command.

<file title> defines the file to be printed or punched. If <file title> is omitted, the workfile is assumed. <file title> can be any file to which the user is allowed access.

If <directory name>/= is specified, each file under the given directory is written.

<sequence range list> defines the portions of the file to be output. When multiple sequence ranges are specified, a blank record separates each range from its successor. If <sequence range list> is omitted, the entire file is output.

The TO option defines the destination of the output. PRINTER directs the output to the site printer, and CARDS directs the output to the site card punch. The default, if this option is omitted, is PRINTER.

The (@) option specifies the portion of each line to be output. If <column range> is specified, then only those columns of each line are output. If ALL is specified, all columns are output adjacent in an unformatted form. If the (@) options are omitted, the lines are output in the format described below.

Output files are unlabeled if printed, and punch files are assigned the title of the file or workfile being punched so that the title appears in an EBCDIC label card at the front of the deck. All non-EBCDIC files are translated to EBCDIC for purposes of punching and printing.

Files punched are output in an unformatted form, that is, the punched deck represents a column-by-column image of the file. Lines longer than a card (80 columns) are truncated.

Each line of printer output is formatted as follows:

1. The record sequence number is printed in columns 1 through 8, right-justified, with leading zeros replaced by blanks (psuedo-sequence numbers are supplied for type DATA or CDATE files). If the sequence field of the record contains non-numeric, nonblank characters or is less than or equal to the previous record sequence number, *SEQERR* is printed instead of the sequence field. In this case, the contents of the sequence field can be found in columns 92 through 99.
2. Columns 9 through 12 are blank.
3. Type DATA and CDATE files use the remainder of the print line (columns 13 through 132) for the text of the record (unless NUMBERED is specified). If the text field of the file is longer than the remainder of the print line, then squeezing and/or truncation is performed if the output options so indicate (these options are described in the following paragraphs). If necessary, the text is split and continued on successive lines of output. To use the entire 132-column print line for text, the single output option UNSEQUENCED must be specified.

CANDE REFERENCE MANUAL

4. For file types other than DATA or CDATA, print columns 13 through 87 are used for the text portion of the record. The text field of type COBOL files (columns 7 through 72) is printed in columns 13 through 78, and the identification field (columns 73 through 78) is printed in columns 81 through 88. For other types of files, the text field is printed in columns 13 through 84 (up to 72 characters per line of printer output). If the text field of the file is longer than 72 characters, then squeezing and/or truncation can be specified. If necessary, the text is split and continued on successive lines of output.
5. Columns 89 through 92 are blank.
6. The sequence field of the record is printed in columns 93 through 100. If the file is of type DATA or CDATA, these columns are blank.
7. Columns 101 through 102 are blank.
8. Columns 103 through 112 are left blank for most types of files; however, for type ALGOL, DCALGOL, and ESPOL files that have patch information in columns 81 through 90, the information is printed in these columns.
9. Columns 113 through 114 are blank.
10. If the NUMBERED option is specified, the relative record number is printed in columns 115 through 120 [prefixed by a pound sign (#)].
11. The remainder of the print line is blank.

Several output options are available for the line printer but are ignored for punched card output. The options are described in the following paragraphs.

COPIES

The COPIES option allows more than one copy to be written at a time. If <directory name>/= is specified, each file under the directory is written the specified number of times.

DOUBLE

The DOUBLE option causes listings to be doubled-spaced.

FORMID

The FORMID option specifies a <string> to be associated with a printer or card punch. The <string> specified may be up to 15 characters long and is displayed on the ODT. (Refer to the discussion of the FORMID attribute in Section 3 of the B 5000/B 6000/B 7000 Series I/O Subsystem Reference Manual.)

CANDE REFERENCE MANUAL

HEXPPLICIT

The HEXPLICIT option causes a record that contains nongraphic characters to be printed on two lines. The two characters representing the hexadecimal notation of the nongraphic character appear vertically. The first HEX character appears on the first line, and the second HEX character appears on the second line immediately below the first. Upper-case letters and other graphic characters appear on the first line with a blank on the second line. Lower-case letters are printed in the upper-case equivalent with a hyphen underscore. A blank line separates each record. The DOUBLE option is ignored if the HEXPLICIT option is used.

NUMBERED

The NUMBERED option causes the number of each record to be printed in columns 115 through 120.

SQUASHED

The SQUASHED option causes any group of multiple blanks to be reduced to a single blank.

TRAINID

The TRAINID option is used to specify which train is to be used on the the line printer when printing this file. (Refer to the discussion of the TRAINID attribute in Section 3 of the B 5000/B 6000/B 7000 Series I/O Subsystem Reference Manual.)

TRUNCATED

The TRUNCATED option causes the text portion of a record that does not fit in the available space on a line to be truncated to one line and terminated by a "/".

UNSEQUENCED

The UNSEQUENCED option causes the sequence number listed at the beginning of each line to be suppressed and the record text to begin in print column 1.

UPCASED

The UPCASED option causes all lower-case letters to be translated to the upper-case equivalent.

Examples

```
write
#RUNNING 8154
#
WRI TO PR:DOUBLE, UP FILEID @1-10 100,150-200,END
#RUNNING 8156
#
```

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES.

5. CONTROL COMMANDS

In this section, the CANDE control commands are described and illustrated. These commands provide the ability to control and interrogate the operating environment. The commands must be preceded by the station control character, represented as a question mark (?) in this manual.

Control commands are nonqueueable and are performed or rejected immediately. Also, some control commands may be processed in parallel with noncontrol commands.

The standardization of controller key-ins has created synonyms for the following control commands:

Existing Command	3.1 Synonym
?DC	?ID
?DP	?DUMP
?MIXA	?MXA
?PC	?SC
?WI	?SI
?WD	?TD

These synonyms are valid for the 3.1 and subsequent software releases.

?AX**Syntax**
Semantics

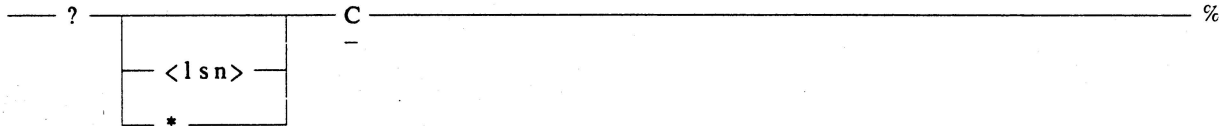
The ?AX command is used to pass text to the currently running program either in response to, or in anticipation of, an ACCEPT statement. If <mix no> is not specified, the currently running program is assumed. If the colon is used, leading blanks are not removed from the <text>. If the colon is omitted, leading blanks are removed before the <text> is presented to the program.

Example

```

GET DON1
#WORKFILE DON1: ALGOL,4 RECORDS,SAVED
#OBJECT FILE PRESENT, SAVED
LIST
100 BEGIN
200 STRING A;
300 ACCEPT(A);
400 END.
#
RUN
#RUNNING 0208
#0208 ACCEPT:.
?AX THIS IS A TEST
#0208 GOING
#ET=54.0 PT=0.0 IO=0.1

```

?C**Syntax****Semantics**

The ?C (completed mix entries) command lists recently completed jobs and tasks.

The syntax and results of this command are controlled to some extent by the ALLMSG CANDE option. The state of the ALLMSG option is controlled by the system operator.

If ALLMSG is RESET, a syntax error is issued if <lsn> or (*) is specified. Only information pertaining to those JOBS or TASKS originating from the user's <lsn> with which his usercode is associated are returned.

If ALLMSG is SET, information about any <lsn> may be requested. The (*) syntax is a shorthand method for specifying the <lsn> to which the user is signed on. If no <lsn> is supplied, all information pertaining to JOBS or TASKS for the user making the request, regardless of origination point, is returned.

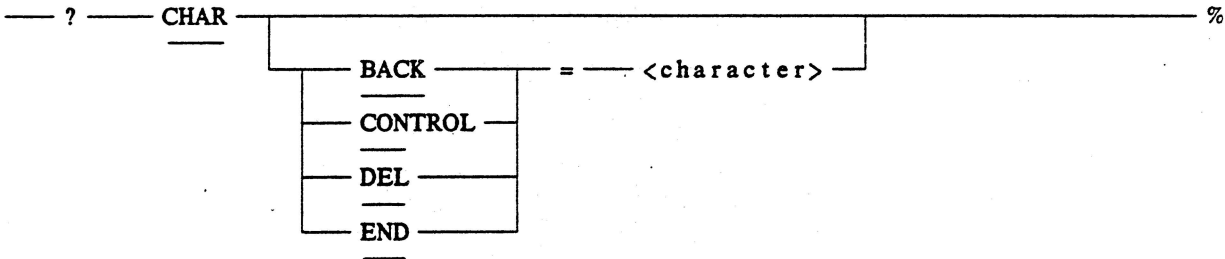
Example

The * in the following example specifies that the entry was completed after the previous display of completed entries on the ODT.

```
?C
----- COMPLETED ENTRIES -----
* 1234/5678 EOJ JOB BACKUP
  1234/5679 EOT SYSTEM/BACKUP ON PACK
  1234/5680 SNTX ALGOL TEST/PROG
  1234/5681 F-DS GEN/TEST/PROG
```

?CHAR

Syntax



Semantics

The ?CHAR command allows four of the characters reserved for special use at a specific terminal to be redefined. The following dynamically alterable characters may be changed during a session:

Backspace	(Teletype Default: left arrow)
Control	(Teletype Default: ?)
Line Delete	(Teletype Default: DEL (Rubout))
End-of-Text	(Teletype Default: CR (Return))

<character> is any special character or ASCII <mnemonic>.

BACK, CONTROL, DEL, and END cause redefinition of the terminal backspace, control, line delete, and end-of-text characters, respectively.

The current state of these four characters may be interrogated by entering ?CHAR without additional text. Characters are represented in the response by graphics, mnemonic names (if graphic representation is not appropriate), or two-digit hexadecimal notation (if no mnemonic exists). Lower-case letters are represented by the upper-case equivalents preceded by a slash.

These reserved characters revert to the NDL default specifications at log-off and Halt/Load time.

Examples

```

?CHAR
#BACK = NUL, CONTROL = ?, DEL = DEL, END = CR
?CHAR CON=*
#BACK = NUL, CONTROL = *, DEL = DEL, END = CR
*CHARACTER BACK=BS
#BACK = BS, CONTROL = *, DEL = DEL, END = CR
  
```

COMMENT

Syntax

```
— ? — % — <text> ————— %
```

Semantics

The COMMENT command treats the <text> as a comment and is intended to be useful in schedule files.

Example

```
?% this is a comment
```

?COUNTS**Syntax**

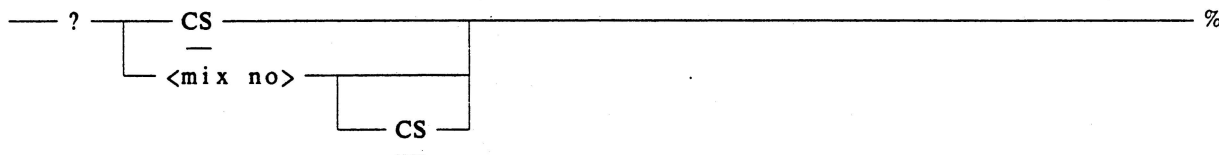
— ? — COUNTS ————— %

Semantics

The ?COUNTS command displays information about the current activity level of CANDE. The response includes the number of stations active, the number of tasks active, the number of workers active, and the number of stations attached.

Example

```
?COUNTS  
# 7 TASKS, 1 WORKER; 33 STATIONS ACTIVE, 78 ATTACHED
```

?CS**Syntax****Semantics**

The ?CS command returns information about the status of a compile task. If <mix no> is not specified, the currently running compile is assumed. The information returned is the sequence number that the compiler is currently processing and the number of syntax errors encountered thus far.

Example

```
?CS
SEQ = 12345678, NO ERRORS
```

?CU**Syntax**

— ? ————— CU ————— %
 └───┬───┘
 <mix no>

Semantics

The ?CU command returns the core utilization for a task. If <mix no> is not specified, the currently running program is assumed.

Example

```
?CU
7228 CORE:  TOTAL   SAVE
           STACK    5191  3787
           SUBSPACE 13860
```


?DENY**Syntax**

— ? — DENY ————— %

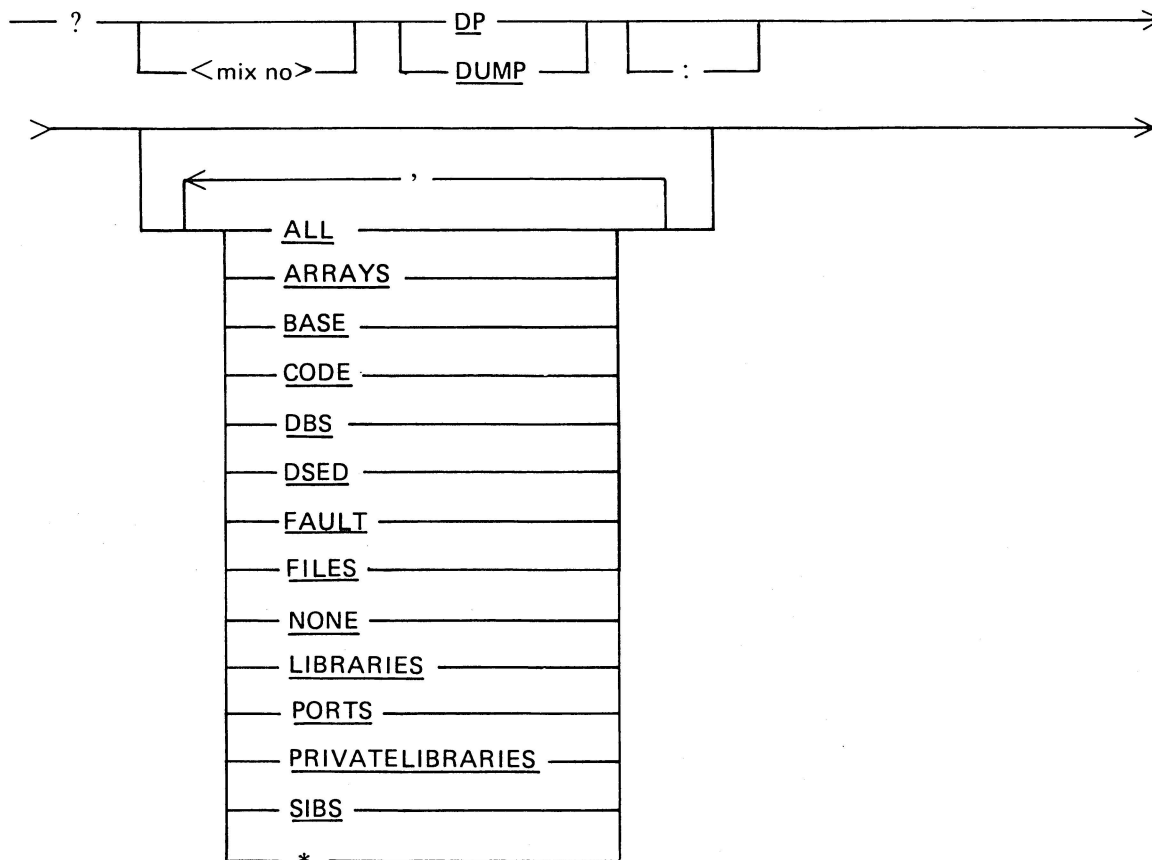
Semantics

The ?DENY command denies program requests for I/O from the terminal. An end-of-file notice is sent to all programs that have files pending, postponed, or open at the terminal.

?END and ?DENY differ in that ?END causes end-of-file action only on the input or I/O file. ?DENY also acts on output files.

Example

```
L DON1
100 BEGIN
200 STRING A;
300 INTEGER I;
400 FILE DCOM(KIND=REMOTE,MYUSE=OUT);
500 A:= "HELLO THERE";
600 DO
700 BEGIN WRITE (DCOM,20,A); WAIT ((3)); I:=I+1; END
800 UNTIL I:=3;
900 END.
#
RUN
#RUNNING 0225
HELLO THERE
?deny
#
#0225 EOF NO LABEL:DCOM @ 003:0000:1
#I-DS @00000700
#ET=12.7 PT=0.1 IO=0.07
R
#RUNNING 5326
HELLO THERE
?END
#
HELLO THERE
HELLO THERE
#ET=21.8 PT=0.1 IO=0.1
```

?DP or ?DUMP**Syntax****Semantics**

The `?DP` command initiates a program dump for the specified task. If `<mix no>` is not specified, the currently running task is assumed.

If one of the options `FAULT` or `DSED` is either explicitly or implicitly (by a `?DP:ALL`) set, a program dump occurs only if the program terminates with fault or is DSeD, respectively. A dump is not taken at the time the command is entered.

Examples

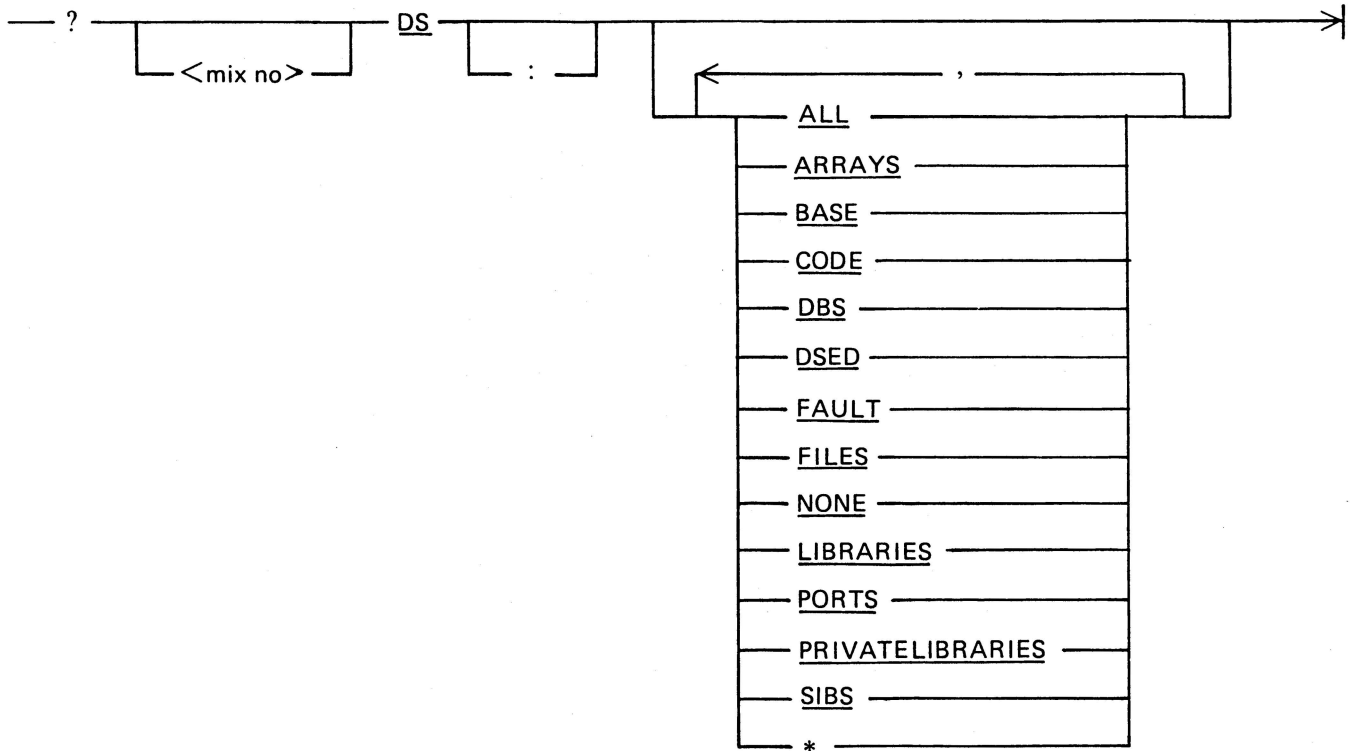
`?DP ARRAYS`

`?DP`

`?DUMP ALL`

?DS

Syntax



Semantics

The `?DS` command terminates the referenced task. If `<mix no>` is not specified, the current CANDE command in progress or the currently running task is terminated. If any program dump options are specified, a program dump is taken prior to termination of the program.

The `?DS` command may also be used to terminate CANDE functions such as `LIST`, `FIND`, `RESEQ`, `WRITE`, and so forth.

Example

```
?DS
#8228 OPERATOR DSED @ 24B:02CF:3*
#O-DS @ 44835200,44904800,00067600,00195600.
#ET=8.7 PT=0.1 IO=0.2
```

ECHO Function

Syntax

— ? ————— %
 └── " — <text> —┘

Semantics

The ECHO function echos <text> back to the terminal. This command may be useful in verifying correct operation of a terminal.

Example

```
? "HELLO DOWN THERE  
"HELLO DOWN THERE
```

?EDIT**Syntax**

— ? — EDIT — <edit specs> ————— %

Semantics

The ?EDIT command allows editing of saved text. The saved text is displayed after it is edited. The EDIT command cannot be used while the station is busy or in sequence mode.

Example

```
100 this is an example.
?edit /an/a good
100 this is a good example.*
#
?repeat
#
list
100 this is a good example.
```

?END**Syntax**

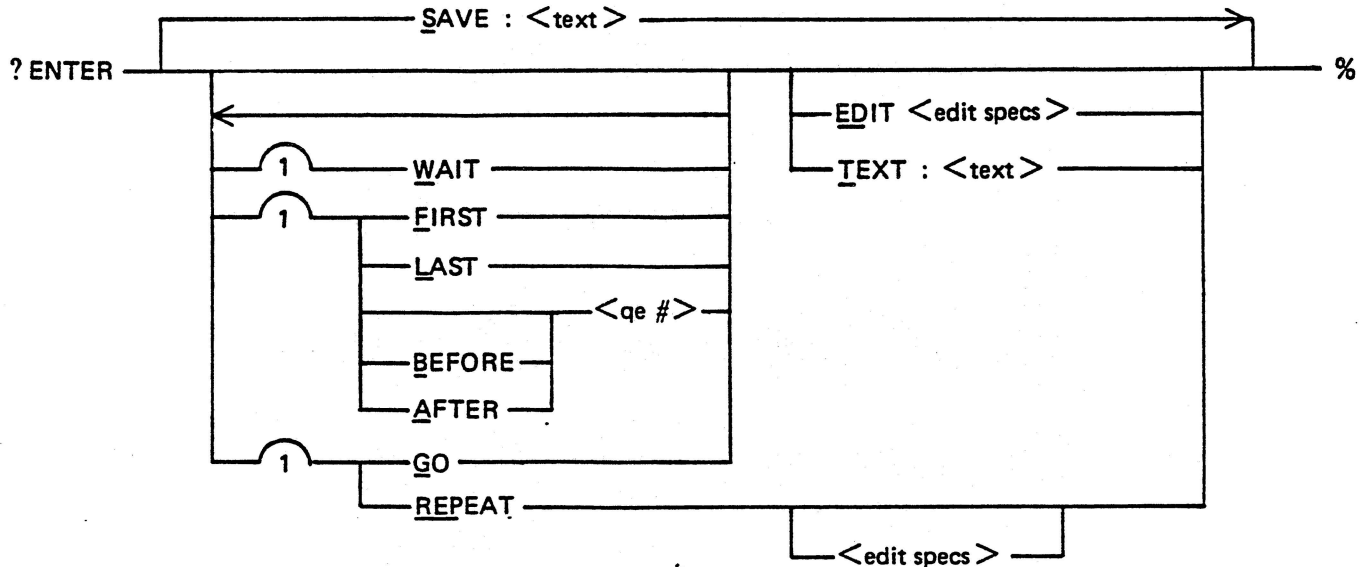
— ? — END ————— %

Semantics

The ?END command signals end-of-file action to the program that has an input or I/O remote file open. (Refer to the DENY control command in this section.)

Example

```
GET DON1
#WORKFILE DON1: ALGOL,4 RECORDS,SAVED
#OBJECT FILE PRESENT,SAVED
LIST
100 BEGIN
200 STRING A;
300 ACCEPT(A);
400 END.
#
run
#RUNNING 0261
0261 ACCEPT:..
?end
#
```

?ENTER**Syntax**

MV2115

Semantics

The `?ENTER` command allows new entries to be placed in the visible portion of the queue or an entry to be made in the saved text queue.

If `SAVE` is specified, the `<text>` following the colon is entered in the saved text queue. If the state of the queued input is pending, the state of the queued input is changed to waiting.

If `SAVE` is not specified, an entry is made in the queue. If no queue entry number (`<qe #>`) is supplied or `FIRST` is specified, the new entry is placed before the entry that is presently first in the queue. `LAST` places the new entry after the entry that is presently last in the queue (possibly overriding visible restrictions). If a `<qe #>` is supplied, the new entry is put `BEFORE`, `AFTER`, or in the place of, the queue entry number specified. The state of the queued input is changed to waiting.

If `WAIT` is specified, the user's queued input state is changed to waiting when that entry becomes the first one in the queue.

If `TEXT` is specified, the `<text>` following the colon constitutes the text of the new queue entry; otherwise, the user's saved text constitutes the text of the new queue entry.

`EDIT` indicates that the saved text is to be edited and displayed before being entered in the queue.

`GO` causes the session to resume with the first entry in the queue. `REPEAT` causes the session to resume with the saved text. If editing specifications follow the `REPEAT` option, the saved text is edited and displayed before being performed.

CANDE REFERENCE MANUAL

The ?ENTER command cannot be used while the station is busy or in sequence mode.

Example

```
E
#UPDATING
#COMPILING 5361
SA; G NEXT
#QUEUED
    WAIT(3);
00000740 ERROR-ILLEGAL EVENT DESIGNATOR
#SNTX
#ET=9.2 PT=0.4 IO=0.3
#QUEUED INPUT PENDING
?SH Q
    (1)SA; G NEXT
?WAIT
#
?ENTER BEFORE 1 TEXT: FIX 740/3/(3)
#
?ENTER AFTER 1 TEXT:E
#
?SH Q
    (1) FIX 740/3/(3)
    (2)E
    (3)SA; G NEXT
?GO
#
#UPDATING
#COMPILING 5371
#ET=6.6 PT=0.1 IO=0.1
#
#WORKFILES TEST SAVED
#NO FILE: NEXT
#
```


?FA**Syntax**

— ? ————— FA ————— <file attribute list> ————— %
 └───┬───┘
 <mix no>

Semantics

The ?FA command allows file attributes for referenced tasks waiting on an RSVP message to be changed. If <mix no> is not specified, the currently running program is assumed. Refer to the B 7000/B 6000 Series Input/Output Subsystem Reference Manual, form 5001779, for more information about file attributes.

Example

```
E
#RUNNING 1234
#1234 NO FILE X ON DISK ELSE PACK(DK)
?FA TITLE=NEW/FILE
#GOING
```


?GO

Syntax

— ? — GO ————— %

Semantics

The ?GO command may only be used when the state of the user's queued input is waiting or pending. This command causes the session to be resumed with the first entry in the queue. The first entry is always removed and performed. The state of the user's queued input is set to normal by the ?GO command.

The ?GO command cannot be used while the station is busy.

Example

```
run;l;
#RUNNING 0395
0395 ACCEPT:
?ds
#0395 OPERATOR DSED @ 003:0002:5*
#O-DS @ 00000300
#ET=2:34.0 PT=0.1 IO=0.1
#QUEUED INPUT PENDING
?go
#
100 BEGIN
200 STRING A;
300 ACCEPT(A);
400 END.
```


?HN**Syntax**

— ? — HN —|

Semantics

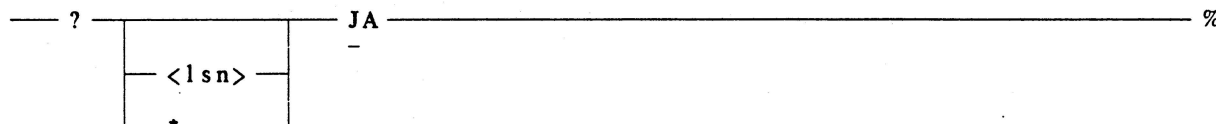
The ?HN (HostName) command displays the current hostname of the host, as well as the hostnames of any other Burroughs Network Architecture (BNA) hosts available for port dialogs or host services functions.

Each host in the network is identified by a hostname which is used to identify the location of resources to which access is desired.

Examples

?HN
#HOSTNAME: BLUE

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR FORMATTING PURPOSES. |

?JA**Syntax****Semantics**

The ?JA command displays the status of jobs running under the usercode of the station. The information about each job or task may include the following:

1. Waiting, scheduled, or error status
2. Job number
3. Mix number
4. Priority
5. Task name
6. Compiler name

The syntax and results of this command are controlled to some extent by the ALLMSG CANDE option. The state of the ALLMSG option is controlled by the system operator.

If ALLMSG is RESET, a syntax error is issued if <lsn> or (*) is specified. Only information pertaining to those JOBS or TASKS originating from the user's <lsn> with which his <usercode> is associated are returned.

If ALLMSG is SET, information about any <lsn> may be requested. The (*) syntax is a shorthand method for specifying the <lsn> to which the user is signed on. If no <lsn> is supplied, all information pertaining to JOBS or TASKS for the user making the request, regardless of origination point, is returned.

Example

```
?JA
----- JOB STRUCTURE -----
5812 JOB 50 RUN MY/PROG ON SIXPACK
W..5820 55 MY/PROG ON SIXPACK
```

?MCS**Syntax**

— ? — MCS — <message control system name> ————— %

Semantics

The ?MCS command causes control of the station to be transferred to another Message Control System (MCS) <message control system name> must designate an NDL-declared MCS name. If an unsaved workfile exists, the following message is displayed:

#REMOVE OR SAVE WORKFILE

The unsaved workfile must then be either removed or saved, and the command must be re-entered.

If a CANDE command is in progress, the following message is displayed:

#STATION IS BUSY

If a remote file is open at the station, the following message is displayed:

#STATION IN USE

However, if the station is busy or in use, the command is discarded.

?MM

Syntax

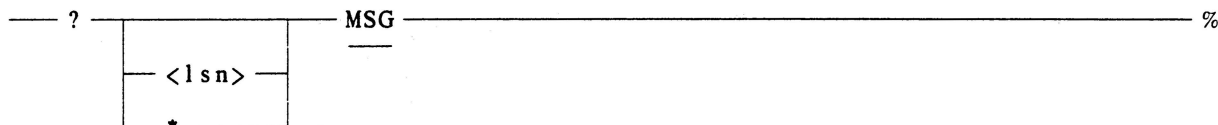
— ? — MM —%

Semantics

The ?MM command displays the status of the memory modules on the system.

Example

```
?mm  
MEMORY STATUS  
16 IN USE 0-15
```

?MSG**Syntax****Semantics**

The ?MSG command displays the most recent messages associated with tasks run under the usercode of the station.

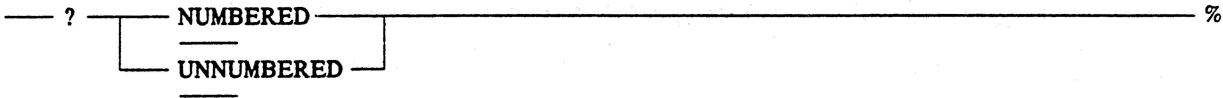
The syntax and results of this command are controlled to some extent by the ALLMSG CANDE option. The state of the ALLMSG option is controlled by the system operator.

If ALLMSG is RESET, a syntax error is issued if <lsn> or (*) is specified. Only information pertaining to those JOBS or TASKS originating from the user s <lsn> with which his <usercode> is associated are returned.

If ALLMSG is SET, information about any <lsn> may be requested. The (*) syntax is a shorthand method for specifying the <lsn> to which the user is signed on. If no <lsn> is supplied, all information pertaining to JOBS or TASKS for the user making the request, regardless of origination point, is returned.

Example

```
?MSG
----- MESSAGES -----
0123 OPERATOR DSED @ 5:001F:5*
0123 OPERATOR DISPLAY: BEGIN TEST ONE
```

?NUMBERED and ?UNNUMBERED**Syntax****Semantics**

The ?NUMBERED and ?UNNUMBERED commands apply only to schedule sessions. If a ?NUMBERED command appears in the input file, the sequence number of each line from the input file appears at the extreme right margin of the image of that line in the output file (unless insufficient room exists on that line). The ?UNNUMBERED command restores the default option; namely, sequence numbers do not appear.

Examples

```

list t
#FILE (UZER)TEST ON USERPACK
100 ?numbered
200 ?report "numbered output"
#
sch
#UPDATING
#SCHEDULE #00009 RUNNING
OUTPUT FILE IS (UZER)SCHOUT/T ON USERPACK
LINE WIDTH IS 72
#
#13:26 FROM SCHEDULE #00009: "NUMBERED OUTPUT"
list schout/t
#FILE (UZER)SCHOUT/T ON USERPACK
#SCHEDULE #00009 COMPLETED
SESSION(S): 1941
#B6800:277 CANDE 30.110; YOU ARE SCHED#000(39)
#SESSION 1941 16:18:50 05/11/78
?numbered
?report "numbered output"
#
#END SESSION 1941 ET=1.7 PT=0.0 IO=0.0
#USER = UZER 16:18:52 05/11/78
#

```

?OF**Syntax**

— ? ———— OF ———— %
 └───┬───┘
 <mix no>

Semantics

The ?OF command may be used in response to a NO FILE message where the file being sought has the attribute OPTIONAL set to TRUE. End-of-file action results from entering this command. If <mix no> is not specified, the currently running program is assumed.

Examples

```
LIST
100 BEGIN
200 FILE F(KIND=DISK,OPTIONAL=TRUE);
300 F.OPEN:=TRUE;
400 END.
#
RUN
#RUNNING 1840
#1840 NO FILE(F) ON PACK(USERFILES)
?OF
#1840 GOING
#ET=7.2 PT=0.3 IO=0.4
```

?OK**Syntax**

— ? ————— OK ————— %
 └───┬───┘
 <mix no>

Semantics

The ?OK command causes processing to be resumed for a suspended task. If <mix no> is not specified, the currently running program is assumed.

Examples

```
?ja
----- JOB STRUCTURE -----
W 1585 JOB 50 WAITTEST

?1585 OK
#1585 GOING
#1585 1585 EOJ JOB WAITTEST
```

?OT**Syntax**

— ? ———— OT ———— <integer> ————— %
 └───┬───┘
 <mix no>

Semantics

The ?OT command is used to interrogate the contents of a specified stack cell in a running program. If <mix no> is not specified, the currently running program is assumed.

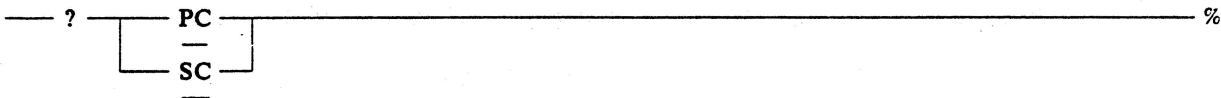
<integer> is the desired offset from the D[2] register.

Examples

```
?ot 3
00096 STACK CELL 3 = 0 00000002B5A (HEX) 11098 (DECIMAL)
```

?PC or ?SC

Syntax



Semantics

The ?PC command displays the current system configuration.

Example

```
?PC  
2 PROCESSORS 1-2  
  D0=02100  
2 MULTIPLEXORS 1-2  
MULTIPLEXOR LOAD LIMIT  
  MPX          LIMIT  TRAFFIC  
  1 (MOD II)   15     N/A  
  2 (MOD II)   15     N/A  
MEMORY STATUS  
16 IN USE 0-15  
PROGRAMMED HALT/LOADS: 0  
STRINGS PRESENT: 1-3
```

?PURGE

Syntax

— ? — PURGE ————— %

Semantics

The ?PURGE command purges all queued input and sets the user's queued input state to normal. Any command in progress is not affected.

Example

```
?ds
#0380 OPERATOR DSED @ 003:0002:5*
#O-DS @ 00000300.
#ET=1:14.4 PT=0.1 IO=0.1
#QUEUED INPUT PENDING
?purge
#QUEUED INPUT PURGED
```

?REPEAT**Syntax**

— ? — REPEAT ————— %
 └───┬───┘
 | |
 └───┬───┘
 <edit specs>

Semantics

The ?REPEAT command causes CANDE to perform the user's saved text as if it were the next line of input. If editing specifications are provided, the text is edited and displayed before being performed. If queued input exists, it is performed when the repeated text finishes. This command cannot be used while the station is busy or in sequence mode.

Example

```
?show all
[2]g don/1*
[1]run*
<0>list*
#
?repeat
#
100 BEGIN
200 STRING A;
300 ACCEPT(A);
400 END.
#
```

?REPORT**Syntax**

— ? — REPORT — <text> ————— %

Semantics

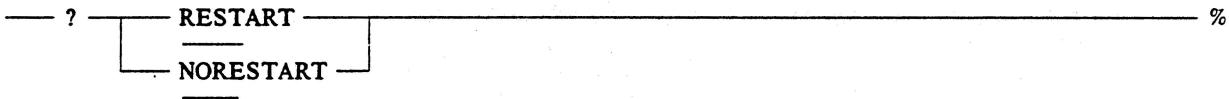
The ?REPORT command is only valid from schedule stations. A command of the form ?REPORT <text> sends the text as a message to any interactive CANDE stations that are logged-on with the same usercode as the schedule session. This command acts as a restricted abbreviation of the ?TO <usercode> <text> command (where the <usercode> is understood). A pound sign (#) acknowledgement appears in the schedule output file if the command is successful; a # NOT ON message appears if no interactive session with that usercode is underway.

Example

```
M DON/3
#WORKFILE DON/3: SEQ
100?REPORT report printed
sch
#UPDATING
#SCHEDULE #00008 RUNNING
  OUTPUT FILE IS (WYMAN)SCHOUT/DON/3 ON USERPACK
  LINE WIDTH IS 72
#
#08:44 FROM SCHEDULE #00008: report printed
```

?RESTART and ?NORESTART

Syntax



Semantics

If a schedule session is aborted because of system failure, the session is restarted after the last ?RESTART command that was encountered prior to the failure. If no ?RESTART command appeared, or if none appeared since a ?NORESTART command, the session is not restarted. If a schedule session is aborted without restart and an unsaved workfile existed, a normal CANDE recovery file is generated. That workfile may be recovered with a RECOVER command in either an interactive or schedule session.

Examples

?REST
?NORE

?RESUME

Syntax

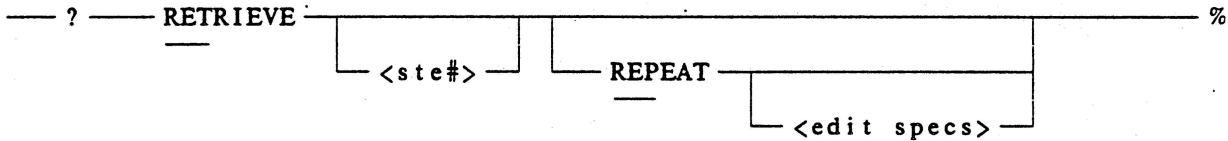
— ? — RESUME ————— %

Semantics

The ?RESUME command is valid only from a schedule station. Any error detected in CANDE syntax, editing, compilation, or abnormal termination of execution causes the remainder of the input lines to be skipped until a ?RESUME command is encountered. If no ?RESUME command is found, the session is aborted. If an unsaved workfile exists at that time, a recovery file is generated and may be recovered in an interactive or schedule session. (An explicit or default BYE, encountered with an unsaved workfile, causes such error action.)

Example

```
list test
#FILE (UZER)TEST ON USERPACK
100 COMPILE TESTIT ALGOL; ALGOL FILE CARD(KIND=DISK,TITLE=DON)
200 ?REPORT "LINE AFTER COMPILE"
300 ?RESUME
400 ?REPORT "LINE AFTER RESUME"
#
list don
#FILE (UZER)DON ON USERPACK
100 BEGIN
200 INTEGER I;
300 J:=1;
400 END.
#
sch
#SCHEDULE #00014 RUNNING
  OUTPUT FILE IS (UZER)SCHOUT/TEST ON USERPACK
  LINE WIDTH IS 72
#
#09:31 FROM SCHEDULE #00014: "LINE AFTER RESUME"
```

?RETRIEVE**Syntax****Semantics**

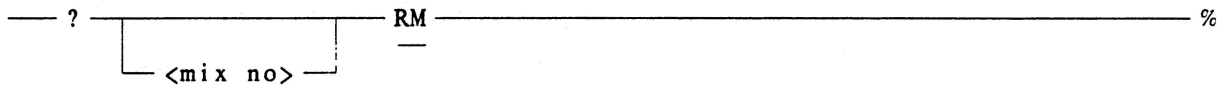
The `?RETRIEVE` command delinks the saved text entry indicated by the saved text entry number `<ste#>` from its current position in the user's saved text queue and makes it the most recent entry. The `<ste#>` specified must be greater than zero and less than or equal to the greatest `<ste#>`. If no `<ste#>` is provided, the number one is assumed.

If `?REPEAT` is specified, the retrieved saved text is performed as the next line of input. If editing specifications are provided, this text is edited and displayed before being performed.

The `?RETRIEVE` command cannot be used while the station is busy or is in sequence mode.

Example

```
?show all
[2]rem sch/don/3*
[1]run*
<0>list*
#
?retrieve 2
#
?show all
[2]run*
[1]list*
<0>rem sch/don/3*
#
```

?RM**Syntax****Semantics**

The ?RM message is used in response to a DUP LIBRARY message. The old file is removed and the new file is retained. If <mix no> is not specified, the currently running program is assumed.

Examples

```
COPY & COMPARE TEST AS TEST/1 FROM USERPACK(PACK) TO USERPACK(PACK)
#RUNNING 1187
#BOT 1188 (UZER)WFLCODE
#BOT 1189 LIBRARY-MAINTENANCE
#1189 (UZER)TEST/1 DUP LIBRARY ON USERPACK PK068 *
?1189 RM
#1189 (UZER)TEST/1 REMOVED ON USERPACK PK068
#1189 (UZER)TEST/1 COPIED.
#EOT 1189 LIBRARY/MAINTENANCE
#EOT 1188 (UZER)WFLCODE
#
```

?SCHEDULE**Syntax**

— ? — SCHEDULE _____ %

Semantics

The ?SCHEDULE control command lists each schedule session currently active or scheduled with the same usercode as that of the station where the command is entered. Whether or not any such sessions exist, a summary of the schedule facility status is provided.

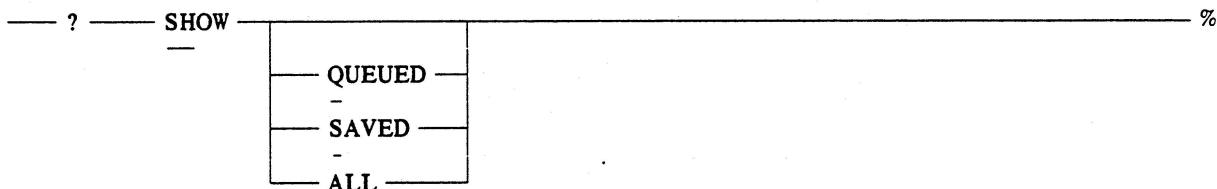
Each schedule session is identified by its schedule number, and an indication of its status is given. The status regarding a schedule session is illustrated below.

"SN=1234"	Active session, logged-on as session #1234
"RUNNING"	Active session, logging-on or off
"WAITING"	Session waiting for schedule station
" 18:30 "	Session to be run after 6:30 pm
" 06:30T"	Session to be run after 6:30 am tomorrow
"(12:00)"	Session to have been run after noon, now waiting
"RESTART"	Session awaiting restart

The summary of the schedule facility status displays (1) the current limit (number of schedule sessions CANDE may run at once); (2) the number of schedule stations attached to CANDE (if different from the limit); (3) the number of active sessions; (4) the number of sessions ready (waiting for a station); (5) the number pending (waiting for an "after" time); and (6) time for the first pending station. Except for the limit, zero numbers are suppressed. The schedule facility is not initialized automatically until one minute after CANDE is started or restarted. In this case, the summary says #SCHEDULE NOT YET INITIALIZED; LIMIT=n (where n is the limit CANDE attempts to establish when initialization takes place).

Example

```
?SCHE
#00020 SN=1123
#SCHEDULE LIMIT=5 NO USERLIMIT ACTIVE=1
```

?SHOW**Syntax****Semantics**

The ?SHOW command causes the saved text queue and/or the visible portion of the input queue to be displayed. Each line displayed is followed by an asterisk (*) to indicate the last character of text. If no modifiers are supplied, the user's saved text is displayed. If QUEUED is specified, only the user's visible queued input is displayed. SAVED displays the user's entire saved text queue, and ALL displays both the saved text queue and the visible queued input.

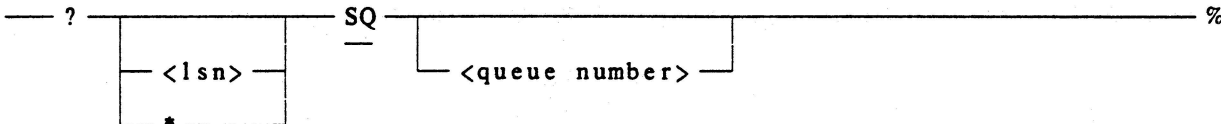
When ALL, QUEUED, and SAVED are specified, each element displayed is preceded by its relative number. The saved text queue entries that are not the most recent are preceded by the queue entry number enclosed in square brackets ([]). The most recent saved text entry is preceded by <0>, and each queued entry is preceded by its queue entry number in parentheses.

A "W" preceding a queued input entry indicates that CANDE is waiting for a user response before taking the queued input entry out of the queue and performing it. (Refer to the ?ENTER queue manipulation command in this section.)

The SHOW command cannot be used while the station is busy or is in sequence mode.

Example

```
?show
sch sch/patch*
?SHOW ALL
[4]list don/1*
[3]g don/1*
[2]?ds*
[1]list*
<0>m don/2*
#
```

?SQ**Syntax****Semantics**

The ?SQ command displays jobs queued to run that were initiated under the usercode of the station. If <queue number> is not specified, entries in all job queues are displayed; otherwise, only entries in the queue referenced by <queue number> are displayed.

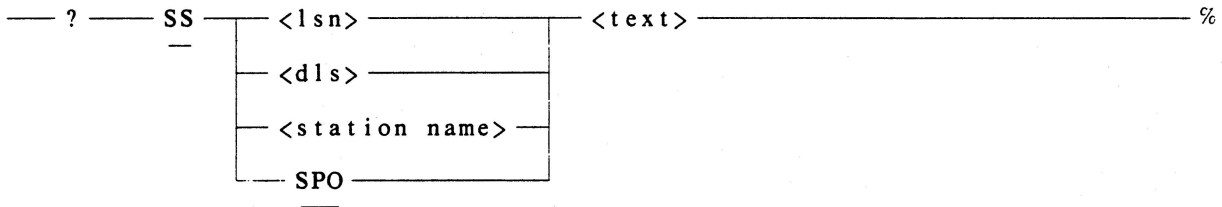
The syntax and results of this command are controlled to some extent by the ALLMSG CANDE option. The state of the ALLMSG option is controlled by the system operator.

If ALLMSG is RESET, a syntax error is issued if <lsn> or (*) is specified. Only information pertaining to those JOBS or TASKS originating from the user's <lsn> with which his <usercode> is associated are returned.

If ALLMSG is SET, information about any <lsn> may be requested. The (*) syntax is a shorthand method for specifying the <lsn> to which the user is signed on. If no <lsn> is supplied, then all information pertaining to JOBS or TASKS for the user making the request, regardless of origination point, is returned.

Example

```
?sq 0
QUEUE 0:
NO ENTRIES
```

?SS**Syntax****Semantics**

The ?SS command is used to send a message to another CANDE station. If SPO is specified, the message is sent to the CANDE SPO. Messages may be sent to the system Operator Display Terminal (ODT) by specifying an <lsn> of 1, which is the ODT.

Examples

```

?ss 1 please mount tape number 92745
#
?SS SPO WHAT IS THE SCHEDULE FOR CANDE SERVICE?
#
?SS 63 Your program does not work
#
  
```

?ST

Syntax

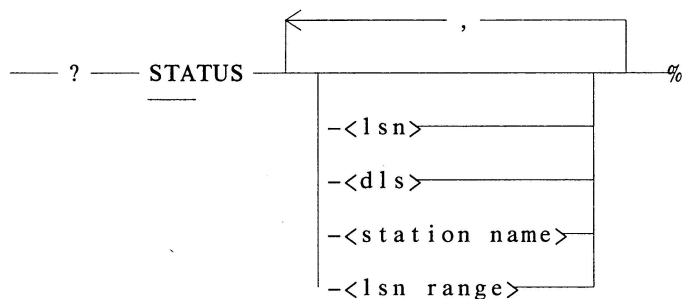
— ? ————— ST ————— %
 └───┬───┘
 <mix no>

Semantics

The ?ST command causes the referenced task to be suspended. If <mix no> is not specified, the currently running program is assumed. A suspended task can be resumed via the ?OK command.

Examples

```
list
100 100 I=1
200 GO TO 100
300 STOP
400 END
run
#RUNNING 1155
?st
#1155 OPERATOR STOPPED
```

?STATUS**Syntax**

<l sn range>

—<low lsn>— - —<high lsn>—|

Semantics

The ?STATUS command is used to display the status of either the currently running task or any terminal on the Data Comm network. If ?STATUS is entered without any other specifications, the status of the current program, the CANDE command (that is, LIST, FIND, REPLACE), or the position of the program in the work, task, or resource queue is displayed.

Examples

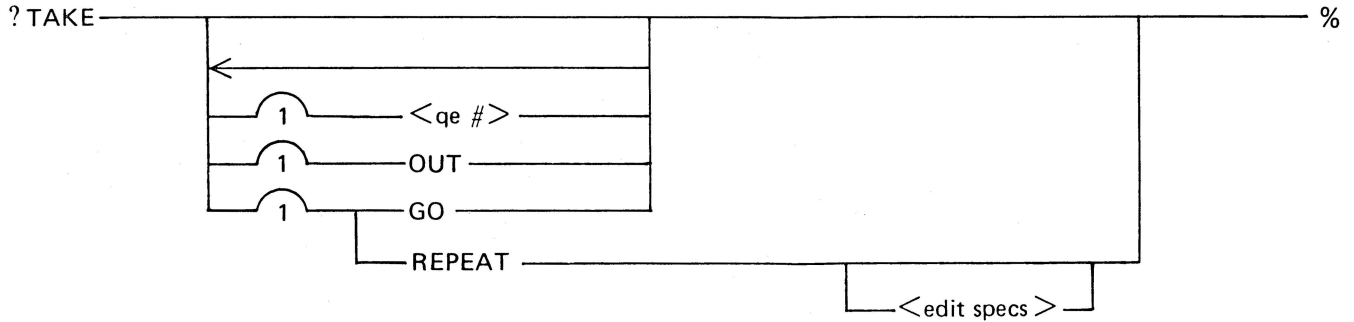
```

?sta
#12:21 ET=19:12:04 PT=0:12:03 IO=0:0:11

?STA 95
SHERRI(95)=0:65:0 RDY ENAB ATT

?sta
#15:55 3RD IN RESOURCE QUEUE

?STA 2,5,9-12,0:10,0:14,TD8441,TTY1
TTY0(2)NOLINE RDY ENAB UNATT
TTY3(5)=0:7:0 RDY ENAB ATT SW=DISCON
TD8441(9)=0:3:1 RDY ENAB ATT
TD0013(10)=0:4:1 RDY ENAB ATT
TD9333(11)=0:9:6 RDY ENAB ATT
TD8450(12)=0:3:3 RDY ENAB ATT
TD81200(60)=0:10:0 RDY ENAB ATT SW=DISCON
TTY1200(61)=0:14:0 UNRDY ENAB ATT
TD8441(9)=0:3:1 RDY ENAB ATT
TTY1(3)=0:5:0 RDY ENAB ATT SW=CONN
```

?TAKE**Syntax****Semantics**

The ?TAKE command takes an entry from the visible portion of the user's queue. If OUT is specified, the entry is thrown away; otherwise, it is entered in the saved text queue of the user. If no queue entry number <qe #> is supplied, the first entry in the queue is taken.

?GO resumes the session with the first entry in the queue.

?REPEAT resumes the session with the saved text. If editing specifications are provided, the saved text is edited and displayed before being performed.

If no entries remain in the queue after completion of the ?TAKE command, the queued input state is set to normal.

The ?TAKE command cannot be used while the station is busy or is in sequence mode.

Example

```
?show
fix 120.A.BCD*
#
?show queued
(1)REM*
(2)make a a*
w(3)1 A/B ON PACK*
(4)100 TEXT FOR A*
#
?take 2
#
?sh
make a a*
#
```

?TF**Syntax**

— ? — TF ————— %

Semantics

The ?TF command displays the value of the memory management parameters.

Example

```
?TF
1) OLAY GOAL = 1% PER MINUTE
2) AVAILMIN = 0%
3) FACTOR = 150%
4) MEM PRIORITY FACTOR = 2%
```

?TI**Syntax**

— ? ————— TI ————— %
 └───┬───┘
 <mix no>

Semantics

The ?TI command displays the process time, elapsed time, and I/O time expended by the referenced task. If <mix no> is not specified, the currently running program is assumed.

Example

```
?TI
TIMES FOR 1238
PROCESS =00:00:08
IO      =00:00:05
ELAPSED =00:23:14
```

?TIME**Syntax**

— ? — TIME ————— %

Semantics

The ?TIME command displays the current time and date.

Example

```
?time  
#10:25 AM TUESDAY, MARCH 29, 1977
```

?TO**Syntax**

— ? — TO — <usercode> — <text> ————— %

Semantics

The ?TO command sends the <text> to all interactive stations where <usercode> is logged-on.

If <usercode> is not logged-on at any interactive station, CANDE responds with #<usercode> NOT ON. Otherwise, CANDE responds with a pound sign (#).

Examples

```
?to purchasing your log sheets are due tomorrow.  
#15:40 FROM PURCHASING ON 11: YOUR LOG SHEETS ARE DUE TOMORROW.  
#
```

?WAIT**Syntax**

— ? — WAIT ————— %

Semantics

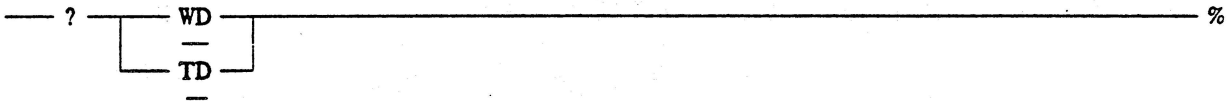
The ?WAIT command sets the state of the user's queued input to waiting. Any command in progress is not affected.

Example

```
E
#RUNNING 5328
SA
#QUEUED
?WAIT
#
#ET=11.2 PT=0.1 IO=0.6
#QUEUED INPUT WAITING
WHAT
#QUEUED, QUEUED INPUT WAITING
?go
#
#WORKFILES ALREADY SAVED
#WORKFILES TEST: ALGOL, 13 RECORDS (THRU 900), SAVED
#OBJECT FILE PRESENT, SAVED
```

?WD or ?TD

Syntax



Semantics

The ?WD command displays the current date.

Example

?WD
DATE IS WEDNESDAY MARCH 30, 1977 (77089)

?WHERE**Syntax**

— ? — WHERE — <usercode> ————— %

Semantics

The ?WHERE command displays the station name and <lsn> of all terminals at which <usercode> is logged-on.

Examples

```
?WH JOHNSON
# JOHNSON ON SHERRI(95)
# JOHNSON ON TTY0(15)
?WH ALAN
# ALAN NOT ON
```

?WHY**Syntax**

_____ ? _____ %
 └───┬───┘ └───┬───┘
 <mix no> WHY
 Y
 -

Semantics

The ?WHY command displays pertinent information about the referenced task. If <mix no> is not specified, the currently running task is assumed. ?Y is a synonym for ?WHY.

Example

```

?WHY
STATUS OF TASK 4226 AT 11:34:15
PRIORITY = 50
ORINATION: <lsn> 96
STACK STATE: HOLDING
SWAP STATUS: ON DISK
PROGRAM: OBJECT/ED ON TIOADMIN
  
```

?WI or ?SI**Syntax**

— ? —

WI
—
SI
—

 ————— %

Semantics

The ?WI command displays the name of the intrinsic file currently in use by the system.

Example

```
?WI
INTRINSICS: SYSTEM/INTRINSICS ON PACK
```

?WM**Syntax**

— ? — WM ————— %

Semantics

The ?WM command displays information about the MCP currently running on the system.

Example

```
?wm
MCP: SYSTEM/MCP II.9.335
COMPILETIME OPTIONS ARE:
  LINEINFO
  MTBF
  LOCKTRACE
  REVERSEPAPERTAPE
H/L REASON: MANUAL
H/L UNIT: 33
SYSTEM SERIAL NO: 277
CATALOG LEVEL: 0
NEXT MCP: NOT SPECIFIED
```

?WRU**Syntax**

_____ <station wru character> _____ %
| |
| ? _____ WRU _____ |
| _____ |

Semantics

The ?WRU command displays identification of the station as well as the version of CANDE that is running. If a user is logged-on at the station, the session number and usercode are displayed. If a user is logged-on and has a charge code, the charge code is also displayed.

Example

```
?WRU  
#B6700:277 CANDE 29.81; YOU ARE SHERRI(95)  
#SESSION = 0942 USER = JOHNSON
```

?WS

Syntax

— ? — WS ————— %

Semantics

The ?WS message displays the name of the current system supervisor program.

Example

```
?WS  
SUPERVISOR: SYSTEM/SUP ON CONTROL
```

?WT**Syntax**

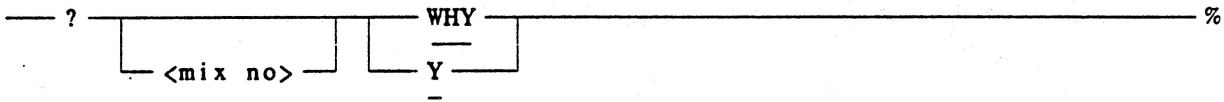
— ? — WT ————— %

Semantics

The ?WT command displays the current time of day.

Example

```
?WT  
TIME IS 09:45:12
```

?Y**Syntax****Semantics**

?Y and ?WHY are synonyms. (Refer to the ?WHY command in this section.)

Example

```
?y
STATUS OF TASK 4226 AT 11:34:15
PRIORITY = 50
ORIGINATION: <lsn> 96
STACK STATE: HOLDING
SWAP STATUS: ON DISK
PROGRAM: OBJECT/ED ON TIOADMIN
```

6. FUNCTIONAL COMMAND GROUPINGS

This section provides groupings of CANDE commands for functional reference purposes.

Task Commands

BIND COMPILE RUN EXECUTE UTILITY WFL START ADD COPY

Library Commands

FILES LFILES REMOVE CHANGE TITLE TYPE SECURITY

Workfile Commands

GET MAKE SAVE RECOVER WHAT

Search Commands

LIST FIND RANGE MATCH

Edit Commands

FIX DELETE MOVE REPLACE RESEQ RENEW INSERT EXCLUDE
TAPE Single-line-entry SEQ

User Information

PASSWORD SO RO CONTINUE FAMILY INDENT MARGIN

Utility Commands

WRITE BDREMOVE DCSTATUS LOG BACKUPPROCESS

Session Commands

SCHEDULE SPLIT BYE HELLO STATUS STOP

System Inquiry Commands

?COUNTS ?DC ?MM ?PC ?STATUS ?TF ?TIME ?WD ?WI
?WM ?WS ?WT

Task Interrogation and Control Commands

?AX ?C ?CS ?CU ?DP ?DS ?FA ?FR ?HI ?JA ?MIXA
?OF ?OK ?OT ?RM ?SQ ?ST ?TI ?WHY

Schedule Control Commands

?% ?REPORT ?NUMBERED ?UNNUMBERED ?RESTART ?NORESTART
?RESUME ?EOL

End-of-File Commands

?END ?DENY

Data Comm Network Commands

?SS ?TO ?WHERE ?WRU

Input Editing

?SHOW ?EDIT ?REPEAT ?RETRIEVE

Queue Manipulation

?GO ?WAIT ?PURGE ?TAKE ?ENTER

CANDE REFERENCE MANUAL

INDEX

ABBREVIATED option	4- 45
ACCESS command	4- 2, 4- 5
<accesscode>	4- 1, 4- 2
ACCESSCODE task attribute	4- 5
ADD command	4- 3
ALLMSG option	5- 3, 5- 23, 5- 25, 5- 27, 5- 42
AN option	4- 48
<any alphanumeric char>	3- 2
<any alphanumeric character>	3-2- A, 3- 3
<any EBCDIC char GEQ hex 40 except quote>	3- 2
<any valid train>	4-116
APASSWORD command	4- 4
?AX	5- 1, 5- 2, 6- 2
BACKUPPROCESS command	4- 6
<base>	3- 1, 4- 41, 4- 59, 4- 75, 4- 88, 4-100
BASIC CONSTRUCTS	1- 1, 2- 5, 4- 51, 4-107
BDREMOVE command	4- 10
BIND command	4- 11
<bridge>	1- 4
BYE command	4- 12, 4- 83
CARRIAGECONTROL option	4- 8
CATALOGUE option	4- 45
CDATA files	4- 26, 4-117
CHANGE	4- 12, 4- 13, 4-105, 6- 1
?CHAR command	5- 4
CHARGE command	4- 14
<chargecode>	4- 13, 4- 14, 4- 82
<column>	4- 53
<column range>	4- 46
COLUMN <start column>	4- 72
COMMENT command	5- 5
COMPARE option	4- 48, 4- 55
COMPILE command	4- 11, 4- 15
<compiler name>	3- 1
<compiler type>	3- 1
COMPILETIME OPTIONS	5- 56
CONTINUE command	4- 17, 4- 27
CONTROL COMMANDS	1- 1, 4-120
COPIES option	4-118
COPY command	4- 19
<copy file list>	4- 19
<count>	4- 31, 4- 72
?COUNTS command	5- 6
CRUNCHED	4- 44
?CS command	5- 7
?CU command	5- 8
<day interval>	4- 95, 4-95- A, 4-95- B
?DC command	5- 9
DCSTATUS command	4- 20
<dd>	4- 95
DELETE command	4- 22
<delim>	4-53- A
<delimiter>	4- 38, 4- 49, 4- 51, 4- 52, 4-53- A
?DENY command	5- 10
<depth>	4- 30, 4- 44, 4- 45
DESTNAME command	4-22- A

CANDE REFERENCE MANUAL

<digit>	3- 5, 4- 95, 4-95- A
DISCARD command	4- 23
<dlm>	3- 2, 4- 31, 4- 32, 4- 34, 4- 35, 4- 36, 4- 72
<dls>	5- 43, 5- 45
DO command	4- 24
DOUBLE option	4- 45, 4-118, 4-119
?DP command	5- 11
?DS command	5- 12
?DUMP	5- 1, 5- 10, 5- 11
ECHO function	5- 13
?EDIT command	5- 14
<edit specs>	3- 2, 4- 34, 5- 14, 5- 34, 5- 38
<end col>	4- 6
<end column>	3- 2, 4- 31, 4- 32, 4- 46, 4- 54, 4- 64, 4- 72, 4-116
?END command	5- 15
?ENTER command	5- 16
<eol>	4- 35
?EOL command	5- 18
EQUAL option	4- 55, 4- 56
EXCLUDE command	4- 26
EXECUTE command	4-22- A, 4- 27
?FA command	5- 19
FAMILY command	4- 28, 4- 83
<family name>	2- 4, 3- 2, 3-2- A, 4- 5, 4- 6, 4- 30, 4-116
FAULT	5- 11
<file attribute list>	5- 19-
<file id>	4- 80
<file identifier>	3- 2, 3-2- A, 3- 8
<file name>	3-2- A, 4- 3, 4- 6, 4- 8, 4- 13, 4- 19, 4- 30, 4- 44, 4- 49, 4- 51, 4- 54, 4- 55, 4- 68, 4- 73, 4- 79, 4- 80, 4- 87, 4-104, 4-105, 4-107, 4-116
<file title>	3- 1, 3-2- A, 4- 24, 4- 26, 4- 27, 4- 31, 4- 32, 4- 38, 4- 41, 4- 46, 4- 47, 4- 49, 4- 51, 4- 57, 4- 58, 4- 64, 4- 65, 4- 75, 4- 77, 4- 79, 4- 86, 4- 95, 4-95- A, 4-110, 4-117
FILEKIND attribute	4-107
<filename>	4- 38, 4- 86
FILEORGANIZATION	4- 44
FILES command	4- 30
FIND command	4- 26, 4- 31, 4- 32, 4- 72
FIX command	3- 2, 4- 34, 4- 35, 4- 36
FLAG option	4- 48
FORMID attribute	4-118
?FR command	5- 20
FUNCTIONAL COMMAND GROUPINGS	1- 1, 4- 1, 5- 60
GENERAL INFORMATION	1- 1, 1- 6
GENERATION	4- 38, 4- 49
GET command	2- 3, 4- 38, 4- 49
?GO command	5- 21
<guard file name>	4- 86
GUARDED option	4- 87
HARDCOPY	4-102
HELLO command	4- 40
HEXPLICIT option	4-118
?HI command	5- 22
<high lsn>	5- 45
<host identifier>	3-2- A
HOSTNAME	5-22- A
<id>	4- 38, 4- 49, 4- 51, 4- 52, 4-53- A
?ID	5- 1, 5- 8, 5- 9

CANDE REFERENCE MANUAL

<identifier> 3-2- B, 3- 3
 <inc> 3- 3, 4- 41, 4- 59, 4- 75, 4- 88, 4-100
 INDENT 6- 1
 <input filetype> 4- 82
 INSERT command 4- 41, 4- 88
 <int filename> 4- 10
 <integer> 3-2- A, 3- 3, 3- 5, 4- 31, 4- 41, 4- 45, 4- 59, 4- 72, 4- 90,
 4- 91, 4- 98, 4-102, 4-116, 5- 31
 INTMODE 4- 44, 4- 45, 4-111
 INTNAME 4-112
 INTRODUCTION 1- 1
 IO option 4- 87

 ?JA command 5- 23
 <julian date> 4- 95, 4-95- A
 JUMP 4- 98

 <keyword> 4- 35
 KIND attribute 2- 4

 LAST 5- 16
 <letter> 3- 5
 LFILES command 4- 43
 <line width> 4- 82, 4- 83
 LIST command 4- 9, 4- 46, 4- 47, 4- 55, 4- 64, 4-101
 LITERAL option 4- 72
 LOAD 4- 39, 4- 48, 4- 49, 5- 32
 LOG command 4- 50
 <loop> 1- 4
 <loops> 1- 2, 1- 3, 1- 4
 <lsn> 2- 5, 4- 20, 4- 96, 5- 3, 5- 22, 5- 23, 5- 24, 5- 25, 5- 27, 5- 42,
 5- 43, 5- 45, 5- 53, 5- 54, 5- 60
 LSN 4- 57, 4- 84
 <lsn range> 5- 45

 MAKE command 2- 3, 4- 51
 MARGIN command 4- 53, 4- 91, 4- 98
 MARKID command 4- 38, 4- 52, 4-53- A
 MATCH command 4- 54, 4- 55
 <match output options> 4- 54
 ?MCS command 5- 24
 MERGE command 4- 58
 <message control system name> 5- 24
 <mix no> 5- 1, 5- 7, 5- 8, 5- 12, 5- 19, 5- 20, 5- 22, 5- 29, 5- 30,
 5- 31, 5- 39, 5- 44, 5- 48, 5- 54, 5- 60
 ?MIXA command 5- 25
 <mm> 4- 95
 ?MM command 5- 26
 <mnemonic> 3- 3, 4- 17, 5- 4
 <modifier> 3- 3, 3- 4, 4- 20, 4- 27, 4- 49, 4- 79, 4-110
 MOVE command 4- 59
 ?MSG command 5- 27
 MULTIPLE COMMANDS 2- 1
 MXA 5- 1, 5- 24

 <n> 4- 36
 <name> 3- 1, 3- 4, 3- 5
 <new directoryname> 4- 13, 4-104, 4-105
 <new file> 4- 54, 4- 55, 4- 56
 <new filename> 4- 13, 4-104, 4-105
 <new password> 4- 4, 4- 62
 <new text> 3- 2, 4- 35, 4- 36, 4- 72
 NEWS command 4- 61
 <nn> 4-22- A

CANDE REFERENCE MANUAL

NOCR	4- 9, 4- 73
NOCRUNCH	4- 6, 4- 8, 4- 31, 4- 32, 4- 73, 4- 79, 4- 80
?NORESTART command	5- 36
<number>	3- 5, 4- 38, 4- 49, 4- 51, 4-53- A, 5- 22
<number of columns>	4- 53
?NUMBERED command	5- 28
NUMBERED option	4-118, 4-119
OBJECT	4- 5, 4- 13, 4- 15, 4- 51, 4- 68, 4- 69, 4- 79, 4- 80, 4- 81, 4- 86, 4-104, 4-105, 4-106, 4-112, 5- 2, 5- 15, 5- 51, 5- 54, 5- 60
<object filename>	4- 15
<object of the loop>	1- 4
?OF command	5- 29
?OK command	5- 30, 5- 44
<old file>	4- 54, 4- 55, 4- 56
<old password>	4- 4, 4- 62
<optional item-1>	1- 4, 1- 5
<optional item-2>	1- 4, 1- 5
<optional items>	1- 2, 1- 3
?OT command	5- 31
OUT option	4- 87
<output filename>	4- 82, 4- 96, 4- 97
<parameters>	4- 95, 4-95- A
<password>	4- 1, 4- 2, 4- 40
PASSWORD command	4- 4, 4- 62
?PC command	5- 32
PRINT	4- 39, 4- 52, 4- 63, 4- 64
PRINTER option	4- 45
PRIVATE option	4- 86
PUBLIC option	4- 87
PUNCH option	4- 47
?PURGE command	5- 33
<queue number>	5- 42
QWAIT option	2- 2, 4- 78, 4- 93
RAILROAD COMPONENTS	1- 3
RAILROAD DIAGRAMS	1- 1
RANGE command	4- 65
Record Formats	3- 8
RECOVER command	2- 4, 4- 67, 4- 80
RECOVERY FILE	2- 3, 2- 4
<recovery number>	4- 23, 4- 67
RECV file	2- 3
REMOTE files	4- 84
REMOVE command	4- 68
RENEW command	4- 70
?REPEAT command	2- 2, 5- 34
REPEAT option	5- 16
REPLACE command	4- 71, 4- 73
?REPORT command	5- 35
<required items>	1- 2, 1- 3
RESEQ command	4- 75
RESPONSES	2- 1
?RESTART command	5- 36
RESULT option	4- 55
?RESUME command	5- 37
?RETRIEVE command	5- 38
<return character>	1- 4
?RM message	5- 39
RMERGE command	4- 77
RO commands	4- 78, 4- 93
RSVP message	5- 25

CANDE REFERENCE MANUAL

RUN 4- 5, 4-22- A, 4-22- B, 4- 27, 4- 78, 4- 79, 4- 80, 4-108, 5- 2,
5- 10, 5- 23, 5- 29, 6- 1

SAVE command 4- 80

SAVED TEXT QUEUE MANIPULATION 2- 1

?SC 5- 1, 5- 31

SCHEDULE command 4- 24, 4- 82, 4- 83, 5- 18

?SCHEDULE control command 5- 40

SECURED option 4- 87

SECURITY command 4- 86

<seq> 4- 34, 4- 53, 4- 72, 4- 90

SEQ command 4- 41, 4- 59, 4- 75, 4- 88

<sequence number> 3-2- A, 3- 5

SEQUENCE option 4- 73

<sequence range> 3- 5, 4- 65, 4- 75

<sequence range list> 3- 5, 4- 6, 4- 8, 4- 22, 4- 24, 4- 26, 4- 31, 4- 32,
4- 38, 4- 41, 4- 46, 4- 49, 4- 54, 4- 55, 4- 58, 4- 59, 4- 64, 4- 65,
4- 70, 4- 73, 4- 77, 4-116, 4-117

?SHOW command 5- 41

?SI 5- 1, 5- 54

SO command 4- 78, 4- 93

<source filetype> 4- 15

<special char> 3- 5, 3- 6, 4- 17, 4- 24, 4- 98

SPLIT command 4- 94, 4-117

SPO 5- 43

?SQ command 5- 42

SQUASHED option 4- 47, 4- 55, 4-119

SQUEEZE option 4- 72

?SS command 5- 43

?ST command 5- 44

<standard compiler> 3- 6, 3- 7, 4- 15

<start col> 4- 6

<start column> 3- 7, 4- 31, 4- 46, 4- 54, 4- 64, 4- 72, 4-116

START command 4-95- A, 4- 96

STARTTIME 4- 95, 4-95- B

<starttime spec> 4- 95, 4-95- A

<starttime specification> 4-95- A

<station name> 4-22- A, 5- 43, 5- 45

<station WRU character> 5- 57

?STATUS command 5- 45

STATUS command 4- 82, 4- 96, 5- 45

STOP command 4- 82, 4- 97

<string> 3- 7, 4- 27, 4- 79, 4-116, 4-118

<sub family> 4- 28

SYNTAX option 4- 15

SYSTEM/DCSTATUS 4- 20, 4- 21

<SYSTEM/DCSTATUS parameters> 4- 20

<SYSTEM/LOGANALYZER parameters> 4- 49, 4- 50

TAB command 4- 98

?TAKE command 5- 46

TAPE command 4-100

<target> 3- 2, 4- 34, 4- 35, 4- 36

<target family> 4- 28

<target text> 4- 72

<targets> 4- 36

<task id> 4- 2, 4- 19

?TD 5- 1, 5- 51

TERM command 4- 83

TERMINAL command 4-102

<text> 4- 31, 4- 53, 4- 90, 4-110, 5- 1, 5- 2, 5- 5, 5- 13, 5- 16, 5- 35,
5- 43, 5- 50

TEXT option 4- 32, 4- 38, 4- 39, 4- 52, 4- 55, 4- 65, 4- 73

?TF command 5- 47

CANDE REFERENCE MANUAL

<this is a null input line>	4- 10
?TI command	5- 48
<time>	4- 12, 4- 40, 4- 82, 4- 83, 4- 95, 4-95- A, 4- 96
?TIME command	5- 49
<time interval>	4- 95, 4-95- A
TITLE attribute	2- 5, 4-104
TITLE command	4- 13, 4-104
TITLE OBJECT	4-105
TITLE SOURCE	4-105
TITLE TO <new filename>	4-104
?TO command	5- 50
TRAINID option	4-119
TRUNCATE option	4- 72
TRUNCATED option	4- 47, 4- 55, 4-119
<type>	3- 1, 3- 7, 4- 51, 4-107
TYPE command	4-107
UNBUFFERED	4-102
?UNNUMBERED commands	5- 28
UPCASED option	4-119
UPDATE command	4- 47, 4-108
USECATALOG option	4- 52
USER IDENTIFICATION	2- 1
<usercode>	3- 1, 3-2- A, 3- 8, 4- 12, 4- 15, 4- 27, 4- 40, 4- 87, 5- 23, 5- 25, 5- 27, 5- 35, 5- 42, 5- 50, 5- 53
USERCODE PASSWORD command	4- 4
UTILITY command	4-110
<volume specification>	4- 2, 4- 19
?WAIT command	5- 51
?WD command	5- 52
WFL command	4- 19, 4-114
WHAT command	4-115
?WHERE command	5- 53
?WHY command	5- 54, 5- 60
?WI command	5- 55
?WM command	5- 56
<work flow language statement(s)>	4-114
<workfilename>	4- 67
WRITE command	4-117
?WRU command	5- 57
?WS message	5- 58
?WT command	5- 59
?Y	5- 22, 5- 59, 5- 60
<yy>	4- 95, 4-95- A
YYDDD	4-95- A
<3-digit index>	4- 10
<7-digit job num>	4- 10
<7-digit task num>	4- 10



