

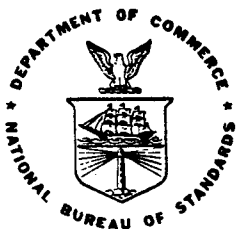
NBSIR 87-3514

SASE - Standards Analysis, Synthesis and Expression Program: USER MANUAL

Steven J. Fenves
May T. Slava
James P. Barnett

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Building Technology
Building Environment Division
Gaithersburg, MD 20899

April 1987



U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

REPRODUCED BY
U.S. DEPARTMENT OF COMMERCE
NATIONAL TECHNICAL
INFORMATION SERVICE
SPRINGFIELD, VA 22161



U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET <i>(See instructions)</i>	1. PUBLICATION OR REPORT NO. NBSIR-87/3514	2. Performing Organ. Report No.	3. Publication Date APRIL 1987
4. TITLE AND SUBTITLE SASE - Standards Analysis, Synthesis and Expression Program: USER MANUAL			
5. AUTHOR(S) Steven J. Fenves; May T. Slava; James P. Barnett			
6. PERFORMING ORGANIZATION <i>(If joint or other than NBS, see instructions)</i> NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		7. Contract/Grant No. 8. Type of Report & Period Covered	
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS <i>(Street, City, State, ZIP)</i> <p style="text-align: center;">NBS Category No. NBS-140</p>			
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
11. ABSTRACT <i>(A 200-word or less-factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i> The Standards Analysis, Synthesis and Expression (SASE) program is intended to assist organizations engaged in the formulation, promulgation and maintenance of standards. It is expected that the user of SASE is familiar with the tutorial material contained in the report <u>Introduction to SASE: Standards, Analysis, Synthesis and Expression.</u> This manual is intended to provide detailed user instructions and serve as the reference manual on the use of the SASE program.			
12. KEY WORDS <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> analysis; building; classification; code; computer; engineering; organization; provisions; scope; specification; standard			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		14. NO. OF PRINTED PAGES 142 15. Price \$18.95	



NBSIR 87-3514

**SASE - STANDARDS ANALYSIS,
SYNTHESIS AND EXPRESSION
PROGRAM: USER MANUAL**

Steven J. Fenves

Department of Civil Engineering
Carnegie-Mellon University

May T. Slava

Department of Civil Engineering
Carnegie-Mellon University

James P. Barnett

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Building Technology
Building Environment Division
Gaithersburg, MD 20899

April 1987

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*



ABSTRACT

The Standards Analysis, Synthesis and Expression (SASE) program is intended to assist organizations engaged in the formulation, promulgation and maintenance of standards.

It is expected that the user of SASE is familiar with the tutorial material contained in the report Introduction to SASE: Standards, Analysis, Synthesis and Expression

This manual is intended to provide detailed user instructions and serve as the reference manual on the use of the SASE program.

Key Words: analysis; building; classification; code; computer; engineering; organization; provisions; scope; specification; standard

TABLE OF CONTENTS

	<u>PAGE</u>
ABSTRACT	iii
DISCLAIMER	x
1 INTERACTIVE CONTROL	1
1.1 General	1
1.1.1 The Usage of "EXIT"	1
1.1.2 How to Run SASE	1
1.2 Command Mode	1
1.2.1 Definition	2
1.2.2 Notation	2
1.2.3 Convention	2
1.2.4 Reducing the Volume of Input	5
1.2.5 System Response	6
1.3 HELP and HELP SYNTAX Mode	7
1.3.1 HELP in Syntax Check	7
2 COMMAND AND FUNCTION DESCRIPTION	8
2.1 Overview	8
2.2 Detailed Description	9
CREATE STANDARD	10
ADD STANDARD	11
DELETE STANDARD	12
DESTROY STANDARD	13
LIST STANDARD	14
MODIFY STANDARD	15
USE STANDARD	16
CREATE VERSION	17
ADD VERSION	18
DELETE VERSION	19
DESTROY VERSION	20
LIST VERSION	21
MODIFY VERSION	22
USE VERSION	23

CREATE CHAPTER	24
ADD CHAPTER	25
DELETE CHAPTER	26
DESTROY CHAPTER	27
LIST CHAPTER	28
MODIFY CHAPTER	29
USE CHAPTER	30
ENTER DATUM	31
ADD DATUM	34
DELETE DATUM	36
DISPLAY DATUM	38
MODIFY DATUM	40
ENTER TABLE	42
ADD TABLE	47
DELETE TABLE	53
DISPLAY TABLE	56
MODIFY TABLE	58
RESEQUENCE TABLE	60
ENTER FUNCTION	62
ADD FUNCTION	63
DELETE FUNCTION	64
DISPLAY FUNCTION	65
MODIFY FUNCTION	67
ENTER CLASSIFIER	68
ADD CLASSIFIER	70
DELETE CLASSIFIER	72

DISPLAY CLASSIFIER	74
MODIFY CLASSIFIER	77
GENERATE TREE	79
DELETE TREE	81
DISPLAY TREE	82
GENERATE NETWORK	83
DELETE NETWORK	85
DISPLAY NETWORK	86
GENERATE HIERARCHY	89
DELETE HIERARCHY	91
DISPLAY HIERARCHY	92
GENERATE SCOPELIST	94
DELETE SCOPELIST	96
DISPLAY SCOPELIST	97
GENERATE INDEX	101
DELETE INDEX	103
DISPLAY INDEX	104
ENTER ORGANIZATION	107
ADD ORGANIZATION	109
CONTINUE ORGANIZATION	110
DELETE ORGANIZATION	111
DISPLAY ORGANIZATION	112
MODIFY ORGANIZATION	114
GENERATE OUTLINE	117
CONTINUE OUTLINE	122
DELETE OUTLINE	124

DISPLAY OUTLINE	125
GENERATE PAGE	127
GENERATE REFERENCE	128
GENERATE SECTION	129
APPENDIX SASE ERROR MESSAGES	130

LIST OF TABLES

	<u>PAGE</u>
Table 1.1 List of Sase Commands	3
Table 1.2 List of Sase Objects	3
Table 1.3 Sase Object and Command Structure	4

LIST OF FIGURES

	<u>PAGE</u>
Figure 3.1 Display of Decision Table 1210	44
Figure 3.2 Display of Decision Table 2130	49
Figure 3.3 Display of Enlarged Decision Table 2130	50
Figure 3.4 Display of Decision Table 1	54
Figure 3.5 Display of Table 1234	60

DISCLAIMER

The program described in this report is furnished by the government and is accepted and used by any recipient with the express understanding that the United States Government makes no warranty, expressed or implied, concerning the accuracy, completeness, reliability, usability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof. This program belongs to the government. Therefore, the recipient further agrees not to assert any proprietary rights therein nor to represent this program to anyone as other than a government program.

CHAPTER 1

INTERACTIVE CONTROL

1.1 General

The central functional component of the SASE system is the interactive control subsystem. It accepts user messages, interprets them, causes the specified functions to be performed, and produces the appropriate output messages. The system generally operates in one major interactive command mode and if desired the user can initiate the help syntax mode from the major mode by typing "HELP SYNTAX"(each input message must be terminated by typing <RET>, where <RET> denotes the "return" or "enter" key on the input keyboard). By typing "EXIT" the user can switch back to the interactive command mode at any time without changing the already generated data.

1.1.1 The Usage of "EXIT"

Entering "EXIT" in any mode will always make the system leave that mode and return to the previous mode. For example: If you are in the syntax help mode, which is called from control mode, typing "EXIT" will lead back to interactive control mode, and typing "EXIT" again will terminate the SASE system and return control to the computer operating system level.

1.1.2 How to Run SASE

The user should first consult the project manager to determine the system "login" and SASE invocation procedures. After the connection between the user terminal and the system is completed, the user can start the SASE package. Entering the system the user must respond to the prompt for terminal line length, this will affect the later SASE display output format. During the session, the appearance of "SASE:" always indicates that the program is ready to accept an input command.

1.2 Command Mode

This is the standard mode of interaction. User input messages are of variable length and may occupy multiple lines on the input terminal. The special character "\$" is needed at the end of each line for continuation, and the system will use " :" as a prompt for continuing lines. The maximum length of input messages cannot exceed 1600 characters which is equivalent to at least 20 lines of continuation. Each input message must be terminated by a <RET>.

1.2.1 Definition

Each input message typically consists of three parts:

- a. COMMAND (as shown in Table 1.1)
- b. OBJECT designation (as shown in Table 1.2)
- c. additional information, consisting of either:
 - (1) input data identified by their data labels; or
 - (2) qualifier, specifying detailed functions to be executed.

The valid COMMAND/OBJECT combinations are shown in Table 1.3. The detailed descriptions of all user commands are given in Chapter 2.

1.2.2 Notation

The notation used in the format throughout Chapter 2 is as follows:

- UPPER CASE: reserved word (the required key part of reserved word is underscored)
- lower case: user input variable
- []: optional part
- || ||: single choice. Use one of the elements enclosed.
- { }: multiple choice. Use one or more of the elements enclosed.
- (): repetition. Use zero, one or more times.

1.2.3 Convention

The convention used for the user input variable types are as follows:

- a. number: a positive integer which consists of at most four digits (e.g., 2314)
- b. name : a string of four characters or less, the first of which must be alphabetical (e.g., A, ABC, A10B, S1.3)
- c. text : any string of characters other than the single quote or apostrophe, enclosed in single quotes. The maximum length of text is 200 characters. A leading blank is not allowed.

Commands

ADD
CREATE
DELETE
DESTROY
DISPLAY
ENTER
GENERATE
LIST
MERGE
MODIFY
RESEQUENCE
USE

Table 1.1 List of Sase Commands

Objects

BOUNDARY
CHAPTER
CLASSIFIER
DATUM
FUNCTION
HIERARCHY
INDEX
NETWORK
ORGANIZATION
OUTLINE
SCOPELIST
STANDARD
TABLE
TREE
VERSION

Table 1.2 List of Sase Objects

COMMAND	CREATION	CHANGE	EXAMINE	REMOVAL	
OBJECTS		R			
		E			
		S			
		E		D	
	C	N	M Q	I	D
	R E	E M	O U	S	E D
	E N	R E	D E	L P	S E
	A T A	A R	I N	U I L	T L
	T E D	T G	F C	S S A	R E
	E R D	E E	Y E	E T Y	O T
				Y E	
DATABASE HEADERS:					
STANDARD	X X	X	X X	X X	
VERSION	X X	X	X X	X X	
CHAPTER	X X	X	X X	X X	
INPUT DATA:					
PAGE		X			
REFERENCE		X			
SECTION		X			
DATUM	X X	X X	X	X	
FUNCTION	X X	X	X	X	
TABLE	X X	X X	X	X	
CLASSIFIER	X X	X X	X	X	
ORGANIZATION	X X	X	X	X	
GENERATED DATA:					
TREE	X X	X	X	X	
NETWORK	X X X	X	X	X	
HIERARCHY	X X	X	X	X	
SCOPELIST	X X	X	X	X	
INDEX	X X	X	X	X	
OUTLINE	X X	X	X	X	

Table 1.3 Sase Object and Command Structure

1.2.4 Reducing the Volume of Input

The system always retains the current COMMAND and OBJECT type designation. These need not be repeated on subsequent entry lines if the same COMMAND and/or OBJECT is entered.

For those optional information noted within "[]", the data labels can be omitted* under the following two conditions:

- a. The user message include both COMMAND and OBJECT, and the label information be entered in the same order as shown in the format.

[Example 1] The user message:
SASE: CRE CHA SEVEN TITLE 'MASONRY'

can be replaced by:

SASE: CRE CHA SEVEN 'MASONRY'

[Example 2] The user message:
SASE: ADD DAT 1200 NAME DESOK P 30 ING 1201,\$
: 1202

can also be entered by:

SASE: ADD DAT 1200 DESOK 30 1201, 1202

* the data labels in one input command have to be all omitted or all given, otherwise a syntax error will occur.

- b. The user messages do not include COMMAND or OBJECT and the user wants to add the same information as the previous command.

[Example 1] The format of adding ingredients to a previously entered datum is defined in Chapter 3 as:

ADD DATUM number INGREDIENTS number (,number)

Thus, the first user input message may be:

SASE: ADD DAT 1234 ING 1239, 1240

To add the same information to succeeding data items, the user needs to enter only:

SASE: 1237 1241, 1242
(meaning: ADD DAT 1237 ING 1241, 1242)

[Example 2] If the user intends to enter a set of classifiers by 2 consecutive commands which have the same kind of header information such as name and parent, then the first classifier may be entered as:

SASE: ENT CLA 101 PARENT 50 NAME AAB

and the succeeding classifier can be entered as:

SASE: 102 40 AAC

(meaning: ENT CLA 102 PAR 40 NA AAC)

1.2.5 System Response

During the command mode the user is prompted from the system by the designator "SASE:" for an input message. The system processes the user input messages through two steps: (1) check if there is any syntax error, (2) if error free, the input will be sent to the DBMS processor to perform the specified function.

After processing, the system responds with one of the following:

- a. An acknowledged message (line feed) and the prompt "SASE:" if no errors are detected and no output is required; or
- b. The desired output and the prompt "SASE:" if no errors are detected and system response is required; or
- c. One or more appropriate error messages (shown in APPENDIX A) and the prompt "SASE:"

The error messages are of two types:

- a. Syntax Error Messages identifying syntax that can not be matched with the format, wrong user input variable types, etc; and
- b. Logic Error Messages relating to particular COMMAND/OBJECT combinations during the implementation in database. At the end of each error message, the object giving rise to this message is indicated by required key word, variable number, or variable name. The error conditions for each user command are listed in Chapter 3.

For some user commands, such as RESEQUENCE TABLE, ENTER ORGANIZATION, ADD ORGANIZATION, MODIFY ORGANIZATION, GENERATE OUTLINE, AND ADD OUTLINE, a dialogue may occur during entry.

The dialogue is preceded by the prompt "D:", "DC:", or "DD:" depending on the commands. At the end of the dialogue, the system will return to the command mode (See Chapter 2 under these user commands).

1.3 HELP and HELP SYNTAX Mode

In SASE, HELP is available for the following three operations:

- Type "HELP HELP" - Explains HELP function.
- Type "HELP EXIT" - Explains EXIT function.
- Type "HELP SYNTAX" - Explains command syntax.

The first two operations produce a general help message. For the third help operation SASE initiates a help syntax mode (in this situation the system prompt changes to "HELP-SYNTAX:"). Then the system prompts for what kind of syntax the user wants help with. Generally, in help syntax mode the following four categories are available: (The instruction specified under each item indicates the method of obtaining HELP.)

1. List all the commands and objects in the system.
"?"
2. For a given command, specifies valid objects.
"command ?"
3. For a given object, specifies valid commands.
"?object"
4. For a given command and object pair, specifies valid qualifiers/data labels.
"command object ?"

1.3.1 HELP in Syntax Check

The user can also type in a question mark ("?") immediately after a syntax error is detected in the user input. The syntax checker will print out the correct label/qualifier that should be used to replace the error input. If the error does not come from a wrong label/qualifier then "NO FURTHER HELP IS AVAILABLE" will be printed.

CHAPTER 2

COMMAND AND FUNCTION DESCRIPTION

2.1 Overview

The user commands can be briefly grouped into three of similar use about how to manipulate the input data.

- I. Database header commands: all commands that are associated with the following OBJECTS. Data created by using these commands are saved in the database permanently.

STANDARD
VERSION
CHAPTER

- II. Input Data Commands: all Commands that associate with the following OBJECTS. Data entered by using these commands are saved in the database permanently.

DATUM
FUNCTION
TABLE
CLASSIFIER
ORGANIZATION

- III. Generated Data Commands: all commands that deal with the following OBJECTS. Based on the input data in the second category, the user can generate different structures by using these commands. The generated data are only available for the SASE session in which they are produced.

TREE
NETWORK
BOUNDARY
HIERARCHY
SCOPELIST
INDEX
OUTLINE

For the user's convenience, the commands are presented in the following sequence:

1. by the order of the OBJECT listed in Table 1-3;
2. within each object type, the COMMAND producing the object (CREATE, ENTER or GENERATE, depending on the object type) is described first;

3. all other commands applicable to the object are alphabetized by the COMMAND.

2.2 Detailed Description

This section provides the detailed description of all the user commands and their functions. The format of each command contains COMMAND/OBJECT/additional data information as described in Section 1.2.1.

Throughout this section, examples are given in upper case. If there is no possible input error and no output, then only the user command example is listed; otherwise, a brief explanation and possible system response in lower case are included.

CREATE STANDARD

FORMAT:

CREATE STANDARD [NAME] name

{	[<u>TITLE</u>]	text	}
	[<u>IDENTIFICATION</u>]	text	
	[<u>COMMENT</u>]	text	

FUNCTION:

The named STANDARD is entered into the SASE directory and is available for subsequent processing.

At least one attribute value for TITLE, IDENTIFICATION, and COMMENT may be entered.

RESTRICTIONS:

1. The STANDARD name must be unique.

EXAMPLES:

1. To CREATE a STANDARD named ATC with COMMENT, TITLE the user may enter each attribute preceded by its label:

```
SASE:  CRE STA ATC COM 'EDITION OF MAY, 1980' T $
        'SEISMIC PROVISION'
```

2. To CREATE a STANDARD named NBS with TITLE, and IDENTIFICATION , the user may enter the attributes without labels in the order shown:

```
SASE:  CRE STA NBS 'STANDARD PROVISION' 'HARRIS'
```

ERRORS:

1. Attempt to CREATE a STANDARD with an existing name.

ADD STANDARD

FORMAT:

ADD STANDARD

{	[<u>T</u> ITLE]	text	}
	[<u>I</u> IDENTIFICATION]	text	
	[<u>C</u> OMMENT]	text	

FUNCTION:

Attributes of the current STANDARD are ADDED.

RESTRICTIONS:

1. A current STANDARD must be defined.
2. An attribute may be ADDED only if it has not been entered before.
3. Each attribute value must be preceded by its label.

EXAMPLES:

1. To ADD a TITLE for the current STANDARD:

SASE: ADD STA T 'SEISMIC PROVISION'

2. To ADD COMMENT for the current STANDARD:

SASE: ADD STA COM 'EDITION OF MAY, 1980'

ERRORS:

1. STANDARD is not defined.
2. Attempt to ADD attributes previously defined.

DELETE STANDARD

FORMAT:

DELETE STANDARD

|| ALL ||
|| (labels) ||

Explanation of Terms:

labels ::= { TITLE
IDENTIFICATION
COMMENT }

FUNCTION:

Attributes of the current STANDARD are DELETED. If the word ALL is used, ALL existing attributes are DELETED. Otherwise, only attributes corresponding to the indicated label(s) are DELETED.

RESTRICTIONS:

1. A current STANDARD must be defined.
2. The attributes corresponding to the label(s) must have been previously defined.

EXAMPLES:

1. To DELETE ALL existing attributes of the current STANDARD:

SASE: DEL STA ALL

2. To DELETE the IDENTIFICATION of the current STANDARD:

SASE: DEL STA IDE

ERRORS:

1. STANDARD is not defined.
2. Attempt to DELETE undefined attributes.

DESTROY STANDARD

FORMAT:

DESTROY STANDARD

FUNCTION:

The current STANDARD is DESTROYed. SASE will prompt the user to confirm the delete operation. If confirmed, the current STANDARD is deleted from the directory. All data associated with STANDARD, such as its VERSIONs, etc., are no longer accessible.

RESTRICTIONS:

1. A current STANDARD must be defined.

EXAMPLES:

1. To DELETE the current STANDARD ST1:

```
SASE: USE STA ST1
(ST1) IS THE CURRENT STANDARD
```

```
SASE: DES STA
** CONFIRM TO DESTROY [CURRENT STA] FROM THE DIRECTORY. [Y/N]? Y
** DESTROY COMPLETED **
```

ERRORS:

1. STANDARD is not defined.

LIST STANDARD

FORMAT:

LIST STANDARD [|| ALL ||
(labels) ||]

Explanation of Terms:

labels ::= { TITLE
IDENTIFICATION
COMMENT }

FUNCTION:

A listing of the STANDARDS currently accessible is generated. If no labels are included, only the STANDARD names are shown; else, the attributes corresponding to the specified label(s) are displayed.

RESTRICTIONS:

1. At least one STANDARD must exist in the directory.

EXAMPLES:

1. If the directory only contains two STANDARDS NBS and ATC, and the user wants to LIST all existing STANDARDS without labels:

SASE: LIS STA

<NBS >

<ATC >

2. To LIST all STANDARDS with a specified TITLE:

SASE: LIS STA T

<NBS > TITLE: PROVISION

<ATC > TITLE: SEISMIC PROVISION

ERRORS:

1. Directory contains no STANDARDS.

MODIFY STANDARD

FORMAT:

MODIFY STANDARD

{	[<u>T</u> ITLE]	text	}
	[<u>I</u> DNENTIFICATION]	text	
	[<u>C</u> OMMENT]	text	

FUNCTION:

Attributes of the current STANDARD are replaced.

RESTRICTIONS:

1. A current STANDARD must be defined.
2. The attributes to be MODIFIED must be previously defined.

EXAMPLES:

1. If the current STANDARD's TITLE is 'WIND LOAD', the following will result in the modification of the attribute:

```
SASE: MOD STA T 'SNOW LOAD'
```

2. If the current STANDARD's has no TITLE, the following error will result in the modification of TITLE

```
SASE: MOD STA T 'BASIC SNOW LOAD'
```

```
?D010 ATTEMPT TO MODIFY INFORMATION THAT IS NOT DEFINED T
```

ERRORS:

1. STANDARD is not defined.
2. Attempt to MODIFY undefined attributes.

USE STANDARD

FORMAT:

USE STANDARD [NAME] name

FUNCTION:

This is typically the first command in a session, except when the STANDARD is first CREATED. The command specifies the STANDARD to which the succeeding commands apply. The command stays in effect until replaced by either CREATE STANDARD or USE STANDARD.

RESTRICTIONS:

1. The named STANDARD must exist in the SASE directory.

EXAMPLES:

1. To specify the current STANDARD to be NBS.

```
SASE: USE STA NA NBS
```

```
** <NBS > IS PUT INTO THE CURRENT STANDARD **
```

ERRORS:

1. Attempt to USE an undefined STANDARD.

CREATE VERSION

FORMAT:

CREATE VERSION [NAME] name

```
|| { [TITLE]      text }           [COPY name2  [IN name1]] ||
|| { [IDENTIFICATION] text }
|| { [COMMENT]    text }
```

name1 - STANDARD name to be copied

name2 - VERSION name to be copied

FUNCTION:

The named VERSION is entered into the SASE directory under the current STANDARD and is available for subsequent processing.

Attribute values for TITLE, IDENTIFICATION, and COMMENT may be entered.

The named VERSION may be COPIed into any other existing STANDARD (i.e., currently inactive).

RESTRICTIONS:

1. The VERSION name must be unique in each STANDARD.

EXAMPLES:

1. To CREATE a VERSION named JU76 with TITLE, the user may enter each attribute preceded by its label:

```
SASE:  CRE VER JU76 T 'JUNE 76 VERSION'
```

```
** <JU76> IS PUT INTO THE CURRENT VERSION **
```

2. To CREATE a VERSION named ABC with TITLE, and IDENTIFICATION, the user may enter the attributes without labels in the order shown:

```
SASE:  CRE VER ABC 'VERSION PROVISION' 'HARRIS'
```

```
** <ABC > IS PUT INTO THE CURRENT VERSION **
```

ERRORS:

1. Attempt to CREATE a VERSION with an existing name.

ADD VERSION

FORMAT:

ADD VERSION

{	[<u>T</u> ITLE]	text	}
	[<u>I</u> DNENTIFICATION]	text	}
	[<u>C</u> OMMENT]	text	}

FUNCTION:

Attributes of the current VERSION are ADDED.

RESTRICTIONS:

1. A current VERSION must be defined.
2. An attribute may be ADDED only if it has not been entered before.
3. Each attribute value must be preceded by its label.

EXAMPLES:

1. To ADD a TITLE for the current VERSION:
SASE: ADD VER T 'SEISMIC PROVISION'
2. To ADD a COMMENT for the current VERSION:
SASE: ADD VER COM 'EDITION of MAY, 1980'

ERRORS:

1. VERSION is not defined.
2. Attempt to ADD attributes previously defined.

DELETE VERSION

FORMAT:

DELETE VERSION || ALL ||
 || (labels) ||

labels ::= { TITLE
 IDENTIFICATION
 COMMENT }

FUNCTION:

Attributes of the current VERSION are DELETED. If the word ALL is used, ALL existing attributes are DELETED. Otherwise, only attributes corresponding to the indicated label(s) are DELETED.

RESTRICTIONS:

1. A current VERSION must be defined.
2. The attributes corresponding to the label(s) must have been previously defined.

EXAMPLES:

1. To DELETE ALL existing attributes of the current VERSION:

SASE: DEL VER ALL

2. To DELETE the IDENTIFICATION of the current VERSION:

SASE: DEL VER IDE

ERRORS:

1. VERSION is not defined.
2. Attempt to DELETE undefined attributes.

DESTROY VERSION

FORMAT:

DESTROY VERSION

FUNCTION:

The current VERSION is DESTROYed. SASE will prompt the user to confirm the delete operation. If confirmed, the current VERSION is deleted from the directory. All data associated with the VERSION, such as its CHAPTERs, CLASSIFIERS, etc., are no longer accessible.

RESTRICTIONS:

1. A current VERSION must be defined.

EXAMPLES:

1. To DESTROY the current VERSION VE1:

SASE: USE VER VE1

** <VE1 > IS PUT INTO THE CURRENT VERSION **

SASE: DEL VER

** CONFIRM TO DESTROY [CURRENT VER] FROM THE DIRECTORY. [Y/N]? Y

** DESTROY COMPLETED **

ERRORS:

1. VERSION is not defined.

LIST VERSION

FORMAT:

LIST VERSION [||| ALL
(labels) |||]

Explanation of Term:

labels ::= { TITLE
IDENTIFICATION
COMMENT }

FUNCTION:

A listing of the VERSIONs currently accessible under the current STANDARD is generated. If no labels are included, only the VERSION names are shown; else, the attributes corresponding to the specified label(s) are displayed.

RESTRICTIONS:

1. At least one VERSION must exist in the current STANDARD.

EXAMPLES:

1. If the current STANDARD only contains two VERSIONS VE1 and VE2, and the user wants to LIST all existing VERSIONS without labels:

SASE: LIS VER

<VE1 >

<VE2 >

2. To LIST all VERSIONS with a specified TITLE:

SASE: LIS VER T

<VE1 > TITLE: PROVISION

<VE2 > TITLE: SEISMIC PROVISION

ERRORS:

1. No VERSIONS have been CREATED

MODIFY VERSION

FORMAT:

MODIFY VERSION

{	[<u>T</u> ITLE]	text	}
	[<u>I</u> DNENTIFICATION]	text	
	[<u>C</u> OMMENT]	text	

FUNCTION:

Attributes of the current VERSION are replaced.

RESTRICTIONS:

1. A current VERSION must be defined.
2. The attributes to be MODIFIED must be previously defined.

EXAMPLES:

1. If the current VERSION's TITLE is 'LOADINGS', the following will result in the modification of the attribute:

```
SASE: MOD VER T 'REVISED LOADS'
```

2. If the current VERSION's has no TITLE, the following error will result from the modification of TITLE:

```
SASE: MOD VER T 'SNOW LOAD'
```

```
?D010 ATTEMPT TO MODIFY INFORMATION THAT IS NOT DEFINED T
```

ERRORS:

1. VERSION is not defined.
2. Attempt to MODIFY undefined attributes.

USE VERSION

FORMAT:

USE VERSION [NAME] name

FUNCTION:

This is typically the second command in a session, except when the VERSION is first CREATED. The command specifies the VERSION to which the succeeding commands apply. The command stays in effect until replaced by either CREATE VERSION or USE VERSION.

RESTRICTIONS:

1. The named VERSION must exist in the current STANDARD.

EXAMPLES:

1. To specify the current VERSION to be TEST:

SASE: USE VER NA TEST

** <TEST> IS PUT INTO THE CURRENT VERSION **

ERRORS:

1. Attempt to USE an undefined VERSION.

CREATE CHAPTER

FORMAT:

CREATE CHAPTER [NAME] name

{	[<u>TITLE</u>]	text
	[<u>IDENTIFICATION</u>]	text
	[<u>COMMENT</u>]	text
}		

FUNCTION:

The named CHAPTER is entered into the SASE directory under the current VERSION and is available for subsequent processing.

Attribute values for TITLE, IDENTIFICATION, and COMMENT may be entered.

RESTRICTIONS:

1. The CHAPTER name(s) must be unique in each CHAPTER.

EXAMPLES:

1. To CREATE a CHAPTER named CHI with TITLE, the user may enter each attribute preceded by its label:

```
SASE: CRE CHA CHI T 'MASONRY'
```

```
** <CHI > IS PUT INTO THE CURRENT CHAPTER **
```

2. To CREATE a CHAPTER named XYZ with TITLE and IDENTIFICATION, the user may enter the attributes without labels in the order shown:

```
SASE: CRE CHA XYZ 'WOODWORK' 'HARRIS'
```

```
** <XYZ > IS PUT INTO THE CURRENT CHAPTER **
```

ERRORS:

1. Attempt to CREATE a CHAPTER with an existing name.

ADD CHAPTER

FORMAT:

ADD CHAPTER

{	[<u>TITLE</u>]	text	}
	[<u>IDENTIFICATION</u>]	text	
	[<u>COMMENT</u>]	text	

FUNCTION:

Attributes of the current CHAPTER are ADDED.

RESTRICTIONS:

1. A current CHAPTER must be defined.
2. An attribute may be ADDED only if it has not been entered before.
3. Each attribute value must be preceded by its label.

EXAMPLES:

1. To ADD a TITLE for the current CHAPTER:
SASE: ADD CHA T 'SEISMIC PROVISION'
2. To ADD a COMMENT for the current CHAPTER:
SASE: ADD CHA COM 'EDITION OF MAY, 1980'

ERRORS:

1. CHAPTER is not defined.
2. Attempt to ADD attributes previously defined.

DELETE CHAPTER

FORMAT:

DELETE CHAPTER || ALL ||
 || (labels) ||

Explanation of Term:

labels ::= { TITLE
 IDENTIFICATION
 COMMENT }

FUNCTION:

Attributes of the current CHAPTER are DELETED. If the word ALL is used, ALL existing attributes are DELETED. Otherwise, only attributes corresponding to the indicated label(s) are DELETED.

RESTRICTIONS:

1. A current CHAPTER must be defined.
2. The attributes corresponding to the label(s) must have been previously defined.

EXAMPLES:

1. To DELETE ALL existing attributes of the current CHAPTER:

SASE: DEL CHA ALL

2. To DELETE the IDENTIFICATION of the current CHAPTER:

SASE: DEL CHA IDE

ERRORS:

1. Chapter is not defined.
2. Attempt to delete undefined information.

DESTROY CHAPTER

FORMAT:

DESTROY CHAPTER

FUNCTION:

The current CHAPTER is DESTROYed. SASE will prompt the user to confirm the delete operation. If confirmed, the current CHAPTER is deleted from the directory. All data associated with CHAPTER, such as DATUM, FUNCTION, TABLE, etc. are no longer accessible.

RESTRICTIONS:

1. A current CHAPTER must be defined.

EXAMPLES:

1. To DESTROY the current CHAPTER CH1:

```
SASE: USE CHA CH1
** <CH1 > IS PUT INTO THE CURRENT CHAPTER **
```

```
SASE: DES CHA
** CONFIRM TO DESTROY [CURRENT CHA] FROM THE DIRECTORY. [Y/N]? Y
```

ERRORS:

1. CHAPTER is not defined.

MODIFY CHAPTER

FORMAT:

MODIFY CHAPTER

{	[<u>T</u> ITLE]	text	}
	[<u>I</u> DNENTIFICATION]	text	
	[<u>C</u> OMMENT]	text	

FUNCTION:

Attributes of the current CHAPTER are replaced.

RESTRICTIONS:

1. A current CHAPTER must be defined.
2. The attributes to be MODIFIED must be previously defined.

EXAMPLES:

1. If the current CHAPTER's TITLE is 'FLOOR LOADS', the following will result in the modification of the attribute:

```
SASE: MOD CHA T 'COLUMN LOADS'
```

2. If the current CHAPTER's has no TITLE, the following error will result from the modification of TITLE:

```
SASE: MOD CHA T 'COLUMN SPACING'
```

```
?D010 ATTEMPT TO MODIFY INFORMATION THAT IS NOT DEFINED T
```

ERRORS:

1. CHAPTER is not defined.
2. Attempt to MODIFY undefined attributes.

USE CHAPTER

FORMAT:

USE CHAPTER [NAME] name

FUNCTION:

This is typically the third command in a session, except when the CHAPTER is first CREATED. The command specifies the CHAPTER to which the succeeding commands apply. The command stays in effect until replaced by either CREATE CHAPTER or USE CHAPTER.

RESTRICTIONS:

1. The named CHAPTER must exist in the current VERSION.

EXAMPLES:

1. To specify the current CHAPTER to CH1:

```
SASE: USE CHA NA CH1
```

```
** <CH1 > IS PUT INTO THE CURRENT CHAPTER **
```

ERRORS:

1. Attempt to USE an undefined CHAPTER.

ENTER DATUM

FORMAT:

ENTER DATUM [REFERENCE] number

[NAME]	name
[TITLE]	text
[SECTION]	name
[PAGE]	number
[VALUE]	NUMERIC BOOLEAN BV
[SOURCE]	INPUT DERIVED
[TYPE]	TABLE FUNCTION
[STATUS]	CLASSIFIED UNCLASSIFIED
[UTILIZATION]	REQUIREMENT DETERMINATION
[INGREDIENTS]	number1 (, number1)
[ARGUMENTS]	usage
[EQUIVALENTS]	number1 (, number1)
[COMMENT]	text

Explanation of Terms:

usage ::= { number2 (, number2)
INDEX number2 (, number2)
OUTLINE number2 (, number2) }

FUNCTION:

The DATUM REFERENCE number is ENTERed in the current CHAPTER. Attribute values for all relevant labels may be ENTERed. SOURCE and STATUS need not be ENTERed; they are always generated.

The user can ask SASE to generate the REFERENCE number when ENTERing DATUM. (See GENERATE REFERENCE for details.)

RESTRICTIONS:

1. The DATUM REFERENCE numbers must be unique in the current VERSION.
2. The default value of SOURCE is INPUT if no INGREDIENTS are ENTERed; otherwise, it is automatically set to DERIVED. An error results if the given value of SOURCE conflicts with its default value.
3. If the TYPE is set to be TABLE or FUNCTION, then the DATUM REFERENCE number is used to ENTER a TABLE or FUNCTION.
4. The default value of STATUS is UNCLASSIFIED if no ARGUMENTS are ENTERed; otherwise, it is automatically set to CLASSIFIED when CLASSIFIERS are ENTERed. An error results if the given value of STATUS conflicts with its default value.
5. The default value of UTILIZATION is DETERMINATION if VALUE is ENTERed as NUMERIC or BV (i.e., boolean vector). There is no default value of UTILIZATION if the VALUE is ENTERed as BOOLEAN. However, if the value of UTILIZATION is ENTERed as REQUIREMENT, the default value of VALUE is BOOLEAN. (Note that DETERMINATION may also be BOOLEAN). Inconsistency between VALUE and UTILIZATION will result in an error.
6. The INGREDIENTS or EQUIVALENTS must be defined in the current VERSION. They cannot duplicate or equal the ENTERed DATUM.
7. The ARGUMENTS must be defined CLASSIFIERS in the current VERSION and they cannot duplicate each other. They may be specified to be used for INDEXing only or OUTLINing only; otherwise, they can be used for both.
8. A maximum of 1250 DATUMS can be ENTERed in a single VERSION.

EXAMPLES:

1. To ENTER a DATUM with REFERENCE number, NAME, SECTION, PAGE, VALUE and UTILIZATION, the user may ENTER each attribute preceded by its label:

```
SASE: ENT DAT 1230 NA DESOK SEC SEC12 P 5 $  
V NU UTI DET
```

2. To ENTER the same DATUM and attributes without labels, the user may ENTER:

```
SASE: ENT DAT 1230 DESOK SEC12 5 NU DET
```

ERRORS

1. Attempt to ENTER a DATUM previously defined.
2. Conflict between VALUE and UTILIZATION.
3. Attempt to use undefined CLASSIFIER REFERENCE numbers as ARGUMENTS. Attempt to use undefined DATUM REFERENCE numbers as INGREDIENTS or EQUIVALENTS.
4. Conflict between given value and default of SOURCE or STATUS.

ADD DATUM

FORMAT:

ADD DATUM [REFERENCE] number

[<u>N</u> AME]	name			
[<u>T</u> ITLE]	text			
[<u>S</u> ECTION]	name			
[<u>P</u> AGE]	number			
[<u>V</u> ALUE]	<table style="border: 1px solid black; padding: 2px;"> <tr><td><u>N</u>UMERIC</td></tr> <tr><td><u>B</u>OOLEAN</td></tr> <tr><td><u>B</u>V</td></tr> </table>	<u>N</u> UMERIC	<u>B</u> OOLEAN	<u>B</u> V
<u>N</u> UMERIC				
<u>B</u> OOLEAN				
<u>B</u> V				
[<u>S</u> OURCE]	<table style="border: 1px solid black; padding: 2px;"> <tr><td><u>I</u>NP</td></tr> <tr><td><u>D</u>ERIVED</td></tr> </table>	<u>I</u> NP	<u>D</u> ERIVED	
<u>I</u> NP				
<u>D</u> ERIVED				
[<u>T</u> YPE]	<table style="border: 1px solid black; padding: 2px;"> <tr><td><u>T</u>ABLE</td></tr> <tr><td><u>F</u>UNCTION</td></tr> </table>	<u>T</u> ABLE	<u>F</u> UNCTION	
<u>T</u> ABLE				
<u>F</u> UNCTION				
[<u>S</u> TATUS]	<table style="border: 1px solid black; padding: 2px;"> <tr><td><u>C</u>LASSIFIED</td></tr> <tr><td><u>U</u>NC</td></tr> </table>	<u>C</u> LASSIFIED	<u>U</u> NC	
<u>C</u> LASSIFIED				
<u>U</u> NC				
[<u>U</u> TILIZATION]	<table style="border: 1px solid black; padding: 2px;"> <tr><td><u>R</u>EQ</td></tr> <tr><td><u>D</u>ETERMINATION</td></tr> </table>	<u>R</u> EQ	<u>D</u> ETERMINATION	
<u>R</u> EQ				
<u>D</u> ETERMINATION				
[<u>I</u> NGREDIENTS]	number (, number)			
[<u>A</u> RGUMENTS]	usage			
[<u>E</u> QUIVALENTS]	number (, number)			
[<u>C</u> OMMENTS]	text			

Explanation of Terms:

usage ::= $\left\{ \begin{array}{l} \text{number (, number)} \\ \underline{\text{INDEX}} \text{ number (, number)} \\ \underline{\text{OUTLINE}} \text{ number (, number)} \end{array} \right\}$

FUNCTION:

Attributes of the specified DATUM are ADDED. Attributes to be ADDED must be preceded by their labels.

SOURCE and STATUS will not be in the list of attribute labels: they are handled automatically.

RESTRICTIONS:

1. The DATUM must exist in the current VERSION.
2. Excluding INGREDIENTS, ARGUMENTS and EQUIVALENTS, attributes can be ADDED only if they have not been previously ENTERed or generated by the system.
3. If INGREDIENTS are ADDED to the DATUM, the value of SOURCE is changed to DERIVED.
4. If ARGUMENTS are ADDED to the DATUM, the value of STATUS is changed to CLASSIFIED.
5. SOURCE and STATUS cannot be ADDED; their values are set automatically (See RESTRICTIONS 2 and 4 of ENTER DATUM).
6. To ADD either VALUE or UTILIZATION, the relations between these two must remain consistent. (See RESTRICTION 5 of ENTER DATUM.)
7. The newly ADDED INGREDIENTS, ARGUMENTS and EQUIVALENTS cannot duplicate any originally defined attributes. (See RESTRICTIONS 6 and 7 of ENTER DATUM.)

EXAMPLES:

1. To ADD a NAME to DATUM 1210:
SASE: ADD DAT 1210 NA PARL
2. To ADD INGREDIENTs to DATUM 1101:
SASE: ADD DAT 1101 ING 345, 458, 1200

ERRORS:

1. Attempt to access an undefined DATUM.
2. Attempt to ADD attributes previously defined.
3. Attempt to use undefined DATUM or CLASSIFIER REFERENCE numbers as INGREDIENTS, ARGUMENTS, or EQUIVALENTS.

DELETE DATUM

FORMAT:

DELETE DATUM [REFERENCE] number

		<u>ALL</u>			
		(labels)			
		{ <u>INGREDIENTS</u> <u>ALL</u>			
		number (, number)			
		{ <u>ARGUMENTS</u> <u>ALL</u>			
		number (, number)			
		{ <u>EQUIVALENTS</u> <u>ALL</u>			
		number (, number)			

Explanation of Terms:

labels ::= { NAME
TITLE
SECTION
PAGE
VALUE
SOURCE
TYPE
STATUS
UTILIZATION
COMMENT }

FUNCTION:

If DELETE DATUM ALL is used, the entire DATUM is DELETED. Prior to the DELETion, the system will prompt the user for confirmation. Otherwise, only the attributes corresponding to the indicated labels are DELETED. To DELETE INGREDIENTS, ARGUMENTS, and/or EQUIVALENTS, enter either the word ALL if ALL are to be DELETED, or the REFERENCE numbers of those to be DELETED.

RESTRICTIONS:

1. The DATUM must exist in the current VERSION.
2. The attribute(s) to be DELETED must be defined.
3. SOURCE and STATUS cannot be directly DELETED. However, the DELETion of all INGREDIENTS or ARGUMENTS changes the values of SOURCE or STATUS to INPUT or UNCLASSIFIED, respectively. (See RESTRICTIONS 2 and 4 of ENTER DATUM.)
4. UTILIZATION cannot be DELETED as long as VALUE remains defined. (See RESTRICTION 5 of ENTER DATUM.)

EXAMPLES:

1. To DELETE the TITLE, VALUE and UTILIZATION of DATUM 1529:
SASE: DEL DAT 1519 T V UTI
2. To DELETE the INGREDIENTs 75 and 76 plus ALL ARGUMENTs of DATUM 456:
SASE: DEL DAT 456 ING 75, 76 ARG ALL
3. To DELETE the entire DATUM 123:
SASE: DEL DAT 123 ALL
** DATUM 123 DELETED **

ERRORS:

1. Attempt to access an undefined DATUM.
2. Attempt to DELETE undefined attribute(s).

DISPLAY DATUM

FORMAT:

DISPLAY DATUM || ALL
 || number || || ALL ||
 || number-number || || (labels) || || [PRINTER]

Explanation of Terms:

labels ::= {
 REFERENCE
 NAME
 TITLE
 SECTION
 PAGE
 VALUE
 SOURCE
 TYPE
 STATUS
 UTILIZATION
 INGREDIENTS
 ARGUMENTS
 EQUIVALENTS
 COMMENT
 }

FUNCTION:

DISPLAY DATUM ALL is used to DISPLAY ALL data items in the current CHAPTER. DISPLAY DATUM number is used to DISPLAY attributes of a specific DATUM. DISPLAY DATUM number-number is used to DISPLAY all data items from the first REFERENCE number to the second REFERENCE number, inclusive.

If the second ALL is used, all existing attributes of the data item(s) are DISPLAYed. Otherwise, only the attributes corresponding to the indicated labels are DISPLAYed.

RESTRICTIONS:

1. In DISPLAY DATUM ALL, at least one DATUM must exist in the current CHAPTER.
2. In DISPLAY DATUM number, the specified DATUM number must exist.
3. In DISPLAY DATUM number-number, at least one DATUM must exist between the numbers given.

EXAMPLES:

1. To DISPLAY the REFERENCE numbers and NAMES of ALL existing data items:

SASE: DIS DAT ALL R NA

REFER	NAME
11	DAT11

REFER	NAME
12	DAT12

REFER	NAME
2222	2222

ERRORS:

1. DATUM is not defined.
2. The DISPLAY requested contains no data items.

MODIFY DATUM

FORMAT:

MODIFY DATUM [REFERENCE] number

[<u>N</u> AME]	name
[<u>T</u> ITLE]	text
[<u>S</u> ECTION]	name
[<u>P</u> AGE]	number
[<u>V</u> ALUE]	<u>N</u> UMERIC <u>B</u> OOLEAM <u>B</u> V
[<u>S</u> OURCE]	<u>I</u> NPOT <u>D</u> ERIVED
[<u>T</u> YPE]	<u>T</u> ABLE <u>F</u> UNCTION
[<u>S</u> TATUS]	<u>C</u> LASSIFIED <u>U</u> NCLASSIFIED
[<u>U</u> TILIZATION]	<u>R</u> EQUIREMENT <u>D</u> ETERMINATION
[<u>I</u> NGREDIENTS]	number number (;number,number)
[<u>A</u> RGUMENTS]	usage
[<u>E</u> QUIVALENTS]	number, number (;number,number)
[<u>C</u> OMMENTS]	text

Explanation of Terms:

usage ::= { number (, number)
 INDEX number (, number)
 OUTLINE number (, number) }

FUNCTION:

Attributes of the specified DATUM are MODIFIED. Any MODIFICATION of INGREDIENTS, ARGUMENTS, or EQUIVALENTS must consist of a pair of REFERENCE numbers: old number, new number. The new number replaces the old (original) number.

RESTRICTIONS:

1. The DATUM must exist in the current VERSION.

2. All attributes previously defined can be MODIFIED, except SOURCE and STATUS.
3. The MODIFICATION of UTILIZATION may depend on VALUE. (See RESTRICTION 5 of ENTER DATUM.)
4. To MODIFY INGREDIENTS, ARGUMENTS, or EQUIVALENTS, they must be defined in the current VERSION. They must not duplicate each other or the existing DATUM.

EXAMPLES:

1. To MODIFY ARGUMENT 11 to 100 and INDEX CLASSIFIER 12 to 200:

SASE: MOD DAT 1234 ARG 11,100 IND 12,200

2. To MODIFY the VALUE and TYPE of DATUM 3839:

SASE: MOD DAT 3839 V B TYP FUN

ERRORS:

1. DATUM is not defined.
2. Attempt to MODIFY undefined attributes.
3. Conflict between VALUE and UTILIZATION of DATUM.
4. Attempt to use undefined DATUM or CLASSIFIER REFERENCE numbers as INGREDIENTS, ARGUMENTS or EQUIVALENTS.
5. Duplication of existing INGREDIENTS, ARGUMENTS or EQUIVALENTS by MODIFICATION.

ENTER TABLE

FORMAT:

ENTER TABLE [REFERENCE] number

```
|| CONDITIONS exp1(; exp1) ACTIONS exp2 (; exp2) ||  
|| RULES exp3 (; exp3) ||  
|| [COMMENT text] END ||
```

where exp1 ::= number1 [\$TUB text] (entries)
 exp2 ::= number2 [\$TUB text] number3 (, number3)
 exp3 ::= number4 (entries) number5

Explanation of Terms:

number1- sequential condition row number
number2- sequential action row number
number3- rule (column) number(s) for this action
number4- sequential rule column number
number5- action (row) number for this rule

entries ::= $\left\{ \begin{array}{c} Y \\ N \\ T \\ F \\ + \\ - \\ . \\ I \end{array} \right\}$

where Y, T -yes or true (equivalent)
 N, F -no or false (equivalent)
 +, - -implied yes or no, respectively
 ., I -immaterial (equivalent)

FUNCTION:

A decision TABLE is entered under the current CHAPTER.

A decision TABLE is composed of CONDITIONS, ACTIONS and RULES. A CONDITION is a logical statement that may have only one of two values: true or false. An ACTION is any expression assigning a value to the DATUM defined by the TABLE. A RULE is a statement that prescribes the set of CONDITIONS

necessary for the specified ACTIONS to be performed. A decision TABLE is a structure for defining a set of related RULES.

ENTER TABLE number CONDITIONS exp1 ACTIONS exp2 is used to ENTER a TABLE row by row (CONDITIONS followed by ACTIONS). Each CONDITION row contains an optional STUB and the row of CONDITION entries. Each ACTION row contains an optional STUB and the row of ACTION entries. Each ACTION entry is specified by the RULE number for which this ACTION is to be executed.

ENTER TABLE number RULES exp3 is used to ENTER a TABLE column by column; each column defining a RULE. Each RULE column consists of a list of CONDITION entries followed by an ACTION number pertaining to that particular RULE.

In both formats, the END qualifier must be used to terminate the ENTERing process. Once the TABLE is ENTERed, SASE performs a consistency check and displays the TABLE.

NOTE 1: In ENTERing TABLES, the character "\$" at the end of each line represents continuation. The character ";" represents the repetition of CONDITIONS, ACTIONS, or RULES.

NOTE 2: The ENTER TABLE command involves several complex steps. Consequently, it seems inevitable that the entry of a TABLE may be interrupted by errors. In order to handle this situation, SASE is designed to store all of the CONDITIONS, ACTIONS or RULES which are ENTERed prior to the first detected error item. If, however, the error occurs in the first CONDITION row (format one) or in the first RULE column (format two), nothing will be stored.

As long as the first CONDITION row or RULE column is ENTERed correctly, a dialogue results in response to any subsequent detection of syntax. The system prints out the appropriate error and confirmation messages. From these, the user can recognize which portion of the TABLE has been accepted and where to proceed. A prompt of SASE appears for subsequent entries.

If additional CONDITIONS, ACTIONS or RULES are to be entered, these subsequent lines must be preceded by the key words, CON, ACT or RUL. Otherwise, the keyword COM for comment or END for termination must be the first word of the subsequent input. The ENTERing process is terminated when no more errors are detected and the END qualifier is ENTERed.

RESTRICTIONS:

1. The TABLE can not be previously ENTERed.
2. The REFERENCE number must correspond to a defined DATUM REFERENCE number in the current CHAPTER. The TYPE of DATUM can be either TABLE or undefined. However, upon ENTERing the TABLE, the TYPE of DATUM is automatically set to TABLE.
3. The TABLE must include at least one CONDITION, one ACTION and one RULE.
4. The CONDITIONS, ACTIONS, or RULES must be ENTERed in a sequential manner beginning with number 1.
5. The TABLE size is limited to 30 CONDITIONS, 30 ACTIONS and 30 RULES.

6. When ENTERing a TABLE row by row:

- * the number of entries in each CONDITION row must be the same.
- * the number of entries (number3; RULE or column number) in each ACTION row can not be greater than the number of entries in a CONDITION row.
- * each entry (number3) in a ACTION row must be distinct, since each RULE can have only one ACTION entry.

7. When ENTERing a TABLE column by column:

- * the number of CONDITION entries in each RULE (column) must be the same.
- * there can only be one ACTION number in each RULE.
- * CONDITION and ACTION STUBS cannot be ENTERed; they must be ADDED later.

EXAMPLES:

The following two examples are presented to illustrate two different entry procedures for the same decision table (shown in Figure 3.1).

Figure 3.1 Display of Decision Table 1210

CONDITION/ACTION	RULE			
	1	2	3	4
1 SEISMIC PERFORMANCE CATEGORY= D	* N	Y	Y	Y
2 BUILDING STAGE = NEW	* .	Y	-	N
3 CONDITION C	* .	-	Y	N
4 CONDITION D	* .	N	N	.

1 CDSLRL = SATISFIED	* X	X	X	X

1. This example illustrates ENTERing the TABLE row by row (format one). (To minimize input, the optional CONDITION STUBS and ACTION STUBS are not ENTERed. They are appended later by using the ADD TABLE command.)

```

SASE:  ENT TAB 1210      CON 1 * N Y Y Y $
      :                  ; 2 * . Y - N $
      :                  ; 3 * . - Y N $
      :                  ; 4 * W N N . $
      :                  ACT 1 * 1,2,3,4 $
      :                  END
  
```

*SC10 CON 4 CONTAINS INVALID ENTRYW
 ** TABLE WAS PARTIALLY ENTERED THROUGH:
 CON 3 * . - Y N
 ** PLEASE ENTER MORE CONDITIONS OR START ENTERING ACTIONS

SASE: CON 4 * . N N . \$
 : ACT 1 * 1, 2, 5, 4 END

*SC34 RULE NUMBER IN ACTION 1 EXCEEDS THE NUMBER OF ENTRIES IN
 CONDITION5

** TABLE WAS PARTIALLY ENTERED THROUGH:
 CON 4 * . N N .
 ** PLEASE ENTER MORE CONDITIONS OR START ENTERING ACTIONS

SASE: ACT 1 * 1, 2, 3, 4 END

RUL	1	2	3	4
C 1	N	Y	Y	Y
C 2	.	Y	-	N
C 3	.	-	Y	N
C 4	.	N	N	.
A 1	X	X	X	X

** TABLE 1210 IS SUCCESSFULLY ENTERED **

2. This example illustrates ENTERING the TABLE column by column
 (format two).

SASE: ENT TAB 1210 RUL 2 Y Y - N * 1 \$
 : ; 3 Y - Y N * 1 \$
 : ; 4 Y N N . * 1 \$
 : END

*SC32 RULE NUMBER 2 IS OUT OF SEQUENCE

** NO TABLE IS ACTIVE IN INPUT BUFFER **

SASE: ENT TAB 1210 RUL 1 N . . . * 1 \$
 : ; 2 Y Y - N * 1 \$
 : ; 3 Y - Y N * 1 \$
 : ; 4 Y N N . N * 1

*SC35 SIZE OF ENTRIES FOR RULE 4 IS NOT EQUAL TO THE
 PREVIOUS ONE

** TABLE WAS PARTIALLY ENTERED THROUGH:
 RUL 3 Y - Y N * 1

** PLEASE ENTER MORE RULES OR TYPE "END" TO STOP
 ENTERING THIS TABLE.

SASE: RUL 4 Y N N . * 1 END

RUL	1	2	3	4
C1	N	Y	Y	Y
C2	.	Y	-	N
C3	.	-	Y	N
C4	.	N	N	.
A1	X	X	X	X

** TABLE 1210 IS SUCCESSFULLY ENTERED **

ERRORS:

1. The TABLE REFERENCE number does not correspond to a defined DATUM REFERENCE number.
2. DATUM's TYPE is not TABLE or undefined.
3. CONDITION, ACTION or RULE numbers do not begin 1 or are not sequential.
4. Number of entries in a RULE is not equal to the previous one.
5. The number of CONDITIONS, ACTIONS or RULEs exceeds 30.
6. The RULE number in any ACTION entry is greater than the total number of entries in CONDITION.

ADD TABLE

FORMAT

```
{  ADD TABLE [REFERENCE] number
   }
   CONDITIONS (number1 [STUB text] )
                 (number1 [STUB text] (entries)
   ACTIONS      (number2 [STUB text] )
                 (number2 [STUB text] number3 (,number3))
   RULES        (number4 (entries) number5)
                 ELSE * number5
   COMMENT      text
   [END]
```

Explanation of Terms:

Please refer to ENTER TABLE.

FUNCTION:

Attributes of the specified TABLE are ADDED. Attributes to be ADDED must be preceded by their labels.

The TABLE size can be increased by ADDING new CONDITIONS, ACTIONS or RULES.

To ADD an ELSE RULE to the TABLE, the user must first GENERATE a TREE by specifying "GEN TRE number ELSE". This enables SASE to verify the existence of any ELSE RULES. If ELSE RULES are detected, the ELSE RULE can then be ADDED to the existing TABLE. ADDING the ELSE RULE does not change the original number RULES.

If any logical errors are detected due to an ADDED CONDITION, ACTION or RULE, the execution of ADD TABLE will not be interrupted. The appropriate error message(s) will be displayed, and ADDING can be continued according to the dialogue described under ENTER TABLE.

If the END qualifier is specified, a consistency check on the entire decision TABLE is performed and the TABLE is displayed.

RESTRICTIONS:

1. The TABLE must exist in the current CHAPTER.
2. COMMENT can not be ADDED if it was previously ENTERed.

3. To ADD data to an existing CONDITION, ACTION, or RULE:

* STUB: must be undefined

* Entries:

CONDITION: the CONDITION row must be completely undefined. The number of ADDED entries in the row must equal the original number of RULES.

ACTION: the ACTION row must be completely undefined. The entries (number3) of the ADDED ACTION row must be defined RULE numbers which have no designated ACTION.

RULE: the RULE column must be completely undefined. The number of ADDED CONDITION entries must equal the original number of CONDITIONS. The ACTION number designated by this RULE can be either defined or undefined. If the designated ACTION is undefined, the number of ACTIONS will increase by one.

4. To ADD new CONDITIONS, ACTIONS or RULES:

* CONDITION:

The new CONDITION numbers must be ADDED in sequence beginning at $m+1$ with increments of 1. (m represents the original number of CONDITIONS.)

The number of entries in each new CONDITION must be equal to the original number of RULES.

* ACTION:

The new ACTION numbers must be ADDED in sequence beginning at $n+1$ with increments of 1. (n represents the original number of ACTIONS.)

The entry(ies) of the ADDED ACTION must be defined RULE number(s) which have no designated ACTION.

* RULE:

- # The new RULE numbers must be ADDED in sequence beginning at p+1 with increments of 1. (p represents the original number of RULES.)
- # The number of entries in each RULE must be equal to the original number of CONDITIONS.
- # Each new RULE can select a defined or undefined ACTION number. If the designated ACTION number is undefined, the number of ACTIONS will increase by one.

EXAMPLES:

The following examples are based on the two decision TABLES shown below.

Figure 3.2 Display of Decision Table 2130

TABLE SIZE: 4 CONDITIONS
2 ACTIONS
3 RULES

RUL	1	2	3
C1	Y	Y	
C2	Y	N	
C3			
C4	N		
A1	X		
A2			X

Figure 3.3 Display of Enlarged Decision Table 2130

TABLE SIZE: 6 CONDITIONS
 4 ACTIONS
 6 RULES

RUL	1	2	3	4	5	6
C1	Y	Y	N	+	-	
C2	Y		N	.	Y	+
C3				.	Y	+
C4	N	.	.	+	I	Y
C5	N	.	N	Y	.	N
C6	N	.	.	+	.	Y
A1	X					
A2		X				
A3					X	
A4				X	X	

1. The following two examples illustrate how to ADD data to a TABLE without changing the TABLE size. Each example is based on the original TABLE shown in Figure 3.2.

a. To ADD data to the undefined CONDITION row 3:

SASE: ADD TAB R 2130 CON 3 * - Y N END

RUL	1	2	3
C1	Y	Y	
C2	Y	N	
C3	-	Y	N
C4	N	.	
A1	X		
A2		X	

* As shown, the entries of CONDITION 3 are ADDED. The result leads to partially defined entries in RULE 2. In this situation, only one command, MODIFY TABLE, can be used to complete RULE 2.

b. To ADD an ELSE RULE to the TABLE:

SASE: ADD TAB 2130 RUL ELSE * 2 END


```

RUL  1 2 3 E
-----
C1   Y  Y
C2   Y  N
C3
C4   N  .
-----
A1   X
A2   X X
-----

```

2. The following three examples illustrate various ways to expand the TABLE size from Figure 3.2 to Figure 3.3.

a. To increase both the number of CONDITION rows and RULE columns, two steps may be implemented. (i.e., enlarge CONDITION size then RULE size, or enlarge RULE size then CONDITION size).

(Step One)

```

SASE: ADD TAB 2130 CON 5 * N . N $
      :                6 * N . . END

```

```

RUL  1 2 3
-----
C1   Y  Y
C2   Y  N
C3
C4   N  .
C5   N . N
C6   N . .
-----
A1   X
A2   X
-----

```

(Step Two)

```

SASE: ADD TAB 2130 RUL 4 N . . + Y + * 4 $
      :                5 + Y Y I . . * 4 $
      :                6 - + + Y N Y * 3 END

```

```

RUL      1 2 3 4 5 6
-----
C1       Y  Y N + -
C2       Y  N . Y +
C3             Y . Y +
C4       N  . + I Y
C5       N . N Y . N
C6       N . . + . Y
-----
A1       X
A2             X
A3             X
A4             X X
-----

```

b. In this example, step one and step two are combined into one ADD TABLE command:

```
SASE: ADD TAB 2130 CON 5 * N . N $
      :                   6 * N . . $
      :                   RUL 4 N . . + Y + * 4 $
      :                   5 + Y Y I . . * 4 $
      :                   6 - + + Y N Y * 3 END
```

* This input will result in the same TABLE shown in Figure 3.3.

c. If the STUBS of CONDITIONS 5, 6 and the STUBS of ACTIONS 3, 4 are given, then the number of CONDITION and ACTION rows can be increased by simply ADDing the STUBS without referring to their entries:

```
SASE: ADD TAB 2130 CON 5 'BUILDING' $
      :                   6 'CAPACITY' $
      :                   ACT 3 'FACTOR = 1.0' $
      :                   4 'FACTOR = 0.5' END
```

RUL	1	2	3
C1	Y	N	
C2	Y	N	
C3			
C4	N	.	
C5			
C6			
A1	X		
A2			X
A3			
A4			

DELETE TABLE

FORMAT:

DELETE TABLE [REFERENCE] number

```
|| ALL || | | | | | | | |
|| CONDITIONS || || STUB || || ALL ||  
|| ACTIONS || || ENTRY || || number (, number) ||  
|| || || ALL || ||  
  
|| RULES || || ALL || ||  
|| || || ELSE || ||  
|| || || number (, number) || ||  
  
|| COMMENT ||  
|| [END] ||
```

FUNCTION:

If DELETE TABLE ALL is used, the entire TABLE is DELETED. Otherwise, only the attributes corresponding to the indicated labels are DELETED.

If DELETE TABLE CONDITION/ACTION ALL is used, ALL of the STUB(s) and entry(ies) of the specified CONDITIONS or ACTIONS are DELETED. Otherwise, only a specific STUB or entry is DELETED. If the second ALL is used, ALL of the existing CONDITIONS or ACTIONS in the TABLE are DELETED. Otherwise, only a specified CONDITION or ACTION is affected.

DELETE TABLE RULE ALL is used to DELETE the entire contents of ALL existing RULES. DELETE TABLE RULE ELSE is used to DELETE the contents of all ELSE RULES. DELETE TABLE RULE number (, number) is used to DELETE the contents of the specified RULE(s).

The size of the original TABLE can be reduced in the following manner:

- * To decrease the number of CONDITIONS, the user must specify "DEL TAB CON ALL number (, number)".
- * to decrease the number of ACTIONS, the user must specify "DEL TAB ACT ALL number (, number)".
- * To decrease the number of RULES, the user must specify "DEL TAB RUL number (, number)".

The number indicated in the above three formats must be the last CONDITION, ACTION, or RULE number in the TABLE. Otherwise, only the contents (ie. STUBs or entries) of the specified attribute will be DELETED with the TABLE size left unchanged.

If the END qualifier is specified, a consistency check is performed and the TABLE is displayed.

RESTRICTIONS:

1. The TABLE must be defined in the current CHAPTER.
2. The attributes to be DELETED must be defined.
3. The DELETion of a CONDITION, ACTION, or RULE must not result in a TABLE with less than one CONDITION, one ACTION, and/or one RULE.

EXAMPLES:

1. To DELETE the entire TABLE 125:

```
SASE: DEL TAB 125 ALL
** DELETION COMPLETED **
```

2. To DELETE the entire contents of RULEs 3 and 7 of TABLE 620:

```
SASE: DEL TAB 620 RUL 3, 7
```

3. To DELETE the STUBs of all CONDITIONs in TABLE 389:

```
SASE: DEL TAB 389 CON S ALL
```

4. If TABLE 1 has six RULEs, three CONDITIONs and two ACTIONs as presented in Figure 3.4, then the following process presents how RULEs 3, 5 and 6 can be DELETED and removed from the TABLE:

Figure 3.4 Display of Decision Table 1

RUL	1	2	3	4	5	6
C1	Y	N	Y	Y	N	N
C2	Y	N	N	Y	Y	N
C3	Y	N	N	N	Y	Y
A1	X	X	X			
A2				X	X	X

```
SASE: DEL TAB 1 RUL 6, 3, 5 END
```

RUL	1	2	3	4
C1	Y	N		Y
C2	Y	N		Y
C3	Y	N		N
A1	X	X		
A2				X

- * The system responds by sorting the RULE numbers in descending order (ie. SASE transforms this to be "DEL TAB 1 RUL 6, 5, 3). As a result, RULEs 6 and 5 are removed entirely, while RULE 3 remains in the TABLE with a null entry. (The contents of RULE 3 were DELETED.)

SASE: RES TAB 1 RUL END
D: 1, 2, 4, 3 (number of rules is 4)

RUL	1	2	3	4
C1	Y	N	Y	
C2	Y	N	Y	
C3	Y	N	N	
A1	X	X		
A2			X	

- * In order to DELETE RULE 3, the user must use the RESEQUENCE TABLE command so as to make RULE 3 the last RULE in the TABLE.

SASE: DEL TAB 1 RUL 4 END

RUL	1	2	3
C1	Y	N	Y
C2	Y	N	Y
C3	Y	N	N
A1	X	X	
A2			X

ERRORS:

1. Attempt to access an undefined TABLE.
2. Attempt to DELETE undefined STUB(s) for a CONDITION(s) or ACTION(s).
3. Attempt to DELETE undefined entry(s) for a CONDITION(s) or RULE(s).
4. Attempt to DELETE an ACTION not specified by any RULE.
5. Attempt to DELETE the only CONDITION, ACTION, or RULE in the TABLE.

DISPLAY TABLE

FORMAT:

DISPLAY TABLE [REFERENCE]

	ALL				{ STUB }	
	number				COMMENT	
	number-number				PRINTER	
					SIZE	

FUNCTION:

DISPLAY TABLE ALL is used to DISPLAY ALL decision TABLEs in the current CHAPTER. DISPLAY TABLE number is used to DISPLAY attributes of a specific TABLE. DISPLAY TABLE number-number is used to DISPLAY all decision TABLEs from the first REFERENCE number to the second REFERENCE number, inclusive.

If the qualifier STUB is used, the STUBS for the CONDITIONS and ACTIONS in the TABLE(s) are also DISPLAYed in the output TABLE.

If the qualifier SIZE is used, the number of existing CONDITIONS, ACTIONS and RULEs is DISPLAYed.

If the qualifier PRINTER is used, the output is sent to a PRINT file for off-line PRINTing. Otherwise, the DISPLAY is produced on-line.

RESTRICTIONS:

1. In DISPLAY TABLE ALL, at least one TABLE must exist in the current CHAPTER.
2. In DISPLAY TABLE number, the specified TABLE number must represent a defined TABLE.
3. In DISPLAY TABLE number-number, at least one TABLE must exist between the numbers given.

EXAMPLES:

1. To DISPLAY a specific TABLE including STUBs and the total SIZE:

SASE: DIS TAB 1210 STUB SIZE

SIZE: 2 CONDITIONS, 2 ACTIONS, 2 RULES

```
DECISION TABLE      1  2  E
-----
1 CONDITION ONE *   Y  N
2 CONDITION TWO *   Y  .
*****
1 ACTION ONE      *   X
2 ACTION TWO      *       X  X
-----
```

NOTE: ELSE RULE DOES NOT COUNT IN RULE SIZE.

ERRORS:

1. Attempt to access an undefined TABLE.
2. No TABLEs exist in the current CHAPTER or between the numbers specified.

MODIFY TABLE

FORMAT:

MODIFY TABLE [REFERENCE] number

{	<u>CONDITIONS</u>		(number1 [<u>STUB</u>] text	
			(number1 [<u>STUB</u>] text] (entries))	
	<u>ACTIONS</u>		(number2 [<u>STUB</u>] text	
			(number2 [<u>STUB</u>] text] number3(,number3))	
	<u>RULES</u>		(number4 (entries) number5)	
		ELSE number5		
	<u>COMMENT</u>		text	
	<u>[END]</u>			

Explanation of Terms:

Please refer to ENTER TABLE.

FUNCTION:

Attributes of the specified TABLE are MODIFIED without changing the TABLE size. (ADD may increase the TABLE size, and DELETE may reduce it.) If the END qualifier is specified, a consistency check on the updated TABLE is performed and a simple output TABLE is DISPLAYed.

RESTRICTIONS:

1. The TABLE must be defined in the current CHAPTER.
2. COMMENT, CONDITION numbers, ACTION numbers or RULE numbers must be previously defined.
3. Specific details concerning the MODIFICATION of CONDITIONs, ACTIONs and RULEs are shown below;

* CONDITIONS:

STUB can be MODIFIED if previously defined.

The number of entries in the CONDITION row being MODIFIED must equal the original number of RULES.

* ACTIONS:

STUB can be MODIFIED if previously defined.

The ACTION entries (number3) must be defined RULES having no designated ACTION.

* RULES:

The number of entries in the RULE column being MODIFIED must equal the original number of CONDITIONS.

The new ACTION number chosen to replace the original one must be previously defined.

EXAMPLES:

1. To MODIFY RULE 1 of TABLE 1234:

SASE: MOD TAB 1234 RUL 1 Y Y N * 1

2. To replace the original STUB of CONDITION 3 of TABLE 6900:

SASE: MOD TAB 6900 CON 3 S 'THIS IS A NEW STUB'

ERRORS:

1. Attempt to access an undefined TABLE.
2. Attempt to MODIFY an undefined STUB of a CONDITION or an ACTION.
3. Attempt to MODIFY undefined attributes.

RESEQUENCE TABLE

FORMAT:

```
RESEQUENCE TABLE [REFERENCE] number || CONDITIONS || [END]
                                         || ACTIONS  ||
                                         || RULES   ||
```

Syntax of Dialogue:

```
User entry: || number (number) ||
             || number ([,] number) ||
```

FUNCTION:

The order of CONDITIONS, ACTIONS, or RULES is RESEQUENCED. SASE first prints the total number of CONDITIONS, ACTIONS or RULES in the specified TABLE. Following this, a special prompt of "D:" appears signaling the user to input his/her new sequence of numbers.

If the qualifier END is used, an updated TABLE is DISPLAYed.

RESTRICTIONS:

1. The TABLE must be defined in the current CHAPTER.
2. The number of entries in the new sequence of numbers must equal the original number of CONDITIONS, ACTIONS, or RULES.

EXAMPLES:

1. To RESEQUENCE the RULES of TABLE 1234 given in Figure 3.5:

Figure 3.5 Display of Table 1234

```
RUL  1  2
-----
C1   Y  N
C2   Y  N
-----
A1   X  X
-----
```

SASE: RES TAB 1234 CON END
D: 2 3 (NUMBER OF CONDITIONS IS 2)
? MISSING NUMBER (IN RESEQUENCING TABLE'S
CONDITION, ACTION, OR RULE)1

** ENTER NEW SEQUENCE NUMBERS SEPARATED BY SPACE OR
COMMA IN A SINGLE LINE.

D: 2, 1

RUL 1 2

C1 N Y

C2 N Y

A1 X X

ERRORS:

1. Attempt to access an undefined TABLE.
2. The new sequence of numbers are ENTERed incorrectly during the dialogue.

ENTER FUNCTION

FORMAT:

```
ENTER FUNCTION [REFERENCE] number
                { [BODY] text   }
                { [COMMENT] text }
```

FUNCTION:

A FUNCTION is ENTERed. The attributes BODY and COMMENT may be ENTERed.

RESTRICTIONS:

1. The DATUM REFERENCE number must exist in the current VERSION.
2. The TYPE of the DATUM must be FUNCTION or undefined. If undefined, the TYPE of the DATUM is set to FUNCTION.

EXAMPLES:

1. To ENTER a FUNCTION with REFERENCE number, BODY and COMMENT, the user may ENTER:

```
SASE:  ENT FUN 1234 BOD 'X=Y+Z' COM $
        'ENTERED in 1981'
```

2. To ENTER the same FUNCTION and attributes without labels, the user may ENTER:

```
SASE:  ENT FUN 1234 'X=Y+Z' 'ENTERED IN 1986'
```

ERRORS:

1. DATUM is not defined.
2. DATUM's TYPE is not FUNCTION or undefined.
3. Attempt to ENTER a FUNCTION previously defined.

ADD FUNCTION

FORMAT:

```
ADD FUNCTION [REFERENCE] number  
           { [BODY] text  }  
           { [COMMENT] text }
```

FUNCTION:

Attributes are ADDED to the specified FUNCTION.

RESTRICTIONS:

1. The FUNCTION must be defined.
2. Attributes to be ADDED must be undefined.

EXAMPLES:

1. To ADD a BODY to the FUNCTION 2500:

```
SASE:  ADD FUN 2500 BOD 'A=B*C'
```

ERRORS:

1. Attempt to access an undefined FUNCTION.
2. Attempt to ADD attributes previously defined.

DELETE FUNCTION

FORMAT:

DELETE FUNCTION [REFERENCE] number

```
|| ALL ||  
|| BODY ||  
|| COMMENT ||
```

FUNCTION:

If DELETE FUNCTION ALL is used, the entire FUNCTION is DELETED. Prior to the DELETion, the system will prompt the user for confirmation. Otherwise, only BODY or COMMENT are DELETED.

RESTRICTIONS:

1. The FUNCTION and the attribute(s) to be DELETED must be defined.

EXAMPLES:

1. To DELETE the entire FUNCTION 678:

```
SASE: DEL FUN 678 ALL
```

2. To DELETE the COMMENT of FUNCTION 110:

```
SASE: DEL FUN 110 COM
```

ERRORS:

1. Attempt to access an undefined FUNCTION.
2. Attempt to DELETE undefined attributes.

DISPLAY FUNCTION

FORMAT:

DISPLAY FUNCTION [REFERENCE] || ALL
 number || || ALL ||
 number-number || || (labels) || [PRINTER]

EXPLANATION OF TERMS:

labels ::= { REFERENCE }
 { BODY }
 { COMMENT }

FUNCTION:

DISPLAY FUNCTION ALL is used to DISPLAY ALL FUNCTIONS in the current CHAPTER. DISPLAY FUNCTION number is used only to DISPLAY a specified FUNCTION. DISPLAY FUNCTION number-number is used for the DISPLAY of ALL FUNCTIONS from the first REFERENCE number to the second REFERENCE number, inclusive.

If the second ALL is used, all existing attributes of the FUNCTION(s) are DISPLAYed. Otherwise, only the attributes corresponding to the indicated labels are DISPLAYed.

RESTRICTIONS:

1. In DISPLAY FUNCTION ALL, at least one FUNCTION must exist in the current CHAPTER.
2. In DISPLAY FUNCTION number, the specified number must be defined.
3. In DISPLAY FUNCTION number-number, at least one FUNCTION must exist between the numbers given.

EXAMPLES:

1. To DISPLAY the REFERENCE numbers and BODies of FUNCTIONs 11 through 13:

SASE: DIS FUN 11-13 R BOD

REFER BODY

11 VOLUM=X*Y*Z

REFER BODY

12 X=Y*Z

REFER BODY

13 DETERMINED BY AGENCY

ERRORS:

1. Attempt to access an undefined FUNCTION.
2. The DISPLAY requested contains no FUNCTION(s).

MODIFY FUNCTION

FORMAT:

MODIFY FUNCTION [REFERENCE] number

{BODY text }
{COMMENT text }

FUNCTION:

Attributes of the specified FUNCTION are MODIFIED.

RESTRICTIONS:

1. The FUNCTION and the attributes to be MODIFIED must be defined.

EXAMPLES:

1. To MODIFY the BODY of FUNCTION 1234:

SASE: MOD FUN 1234 BOD 'REAL OBJECT'

ERRORS:

1. Attempt to access an undefined FUNCTION.
2. Attempt to MODIFY undefined information.

ENTER CLASSIFIER

FORMAT:

ENTER CLASSIFIER [REFERENCE] number

```
[NAME]      name
[TITLE]     text
[TYPE]      | ACTUAL          |
              | TRANSPARENT |
|| [PARENT] number                ||
|| [FOSTER] number [relation] (; number [relation]) ||
[COMMENT]  text
```

Explanation of Terms:

```
relation ::= (PRESENT number (, number))
           (ABSENT number (, number) )
```

FUNCTION:

A CLASSIFIER is ENTERed under the current VERSION. Attribute values for all relevant labels may be ENTERed.

A CLASSIFIER may have either a single PARENT or multiple FOSTER parents. When a single PARENT is ENTERed, the CLASSIFIER belongs to the FACET of the parent. FOSTER parent(s) can only be ENTERed for CLASSIFIERS which are facet ROOTs. A single FOSTER parent is ENTERed as if it were a true PARENT. When multiple FOSTER parents are ENTERed, conditional CONTEXT relation(s) specifying the PRESENCE or ABSENCE of other CLASSIFIERS in the HIERARCHY may also be ENTERed. These CONTEXT relation(s) are used to determine the possible inclusion of the ENTERed CLASSIFIER in a field when the HIERARCHY is GENERATED.

RESTRICTIONS:

1. The CLASSIFIER REFERENCE numbers must be unique in the current VERSION.
2. When PARENT, FOSTER, or a CONTEXT relation is specified, the corresponding numbers must be defined CLASSIFIER REFERENCE numbers in the current VERSION.
3. The default value of TYPE is ACTUAL if a single PARENT is ENTERed; TRANSPARENT if FOSTER parent(s) are ENTERed. Inconsistency between TYPE and PARENT/FOSTER designation will result in an error.

EXAMPLES:

1. To ENTER a CLASSIFIER with REFERENCE number, NAME, TITLE and TYPE:

SASE: ENT CLA 2 NA OBJECT T 'BUILDING' TYP ACT

2. To ENTER a CLASSIFIER 4 having FOSTER parent 1 when CLASSIFIER 10 is ABSENT, and FOSTER parent 4 otherwise:

SASE: ENT CLA 4 FOS 1 ABS 10; 4

3. When ENTERING a CLASSIFIER with REFERENCE number, TITLE, PARENT and TYPE, consistency should be maintained between the TYPE and the PARENT/FOSTER designation:

SASE: ENT CLA 6 T 'PART OF BUILDING' PAR 2 TYP TRA

?D075 CONFLICT BETWEEN CLASSIFIER'S TYPE AND PARENT/FOSTER

ERRORS:

1. Attempt to ENTER a CLASSIFIER previously defined.
2. The REFERENCE number of PARENT, FOSTER parent or classifier in the CONTEXT relation is not defined.
3. Conflict between CLASSIFIER TYPE and the PARENT/FOSTER designation.

ADD CLASSIFIER

FORMAT:

ADD CLASSIFIER [REFERENCE] number

[<u>NAME</u>]	name
[<u>TITLE</u>]	text
[<u>TYPE</u>]	<u>ACTUAL</u>
	<u>TRANSPARENT</u>
[<u>PARENT</u>]	number
[<u>FOSTER</u>]	number [relation] (; number [relation])
[<u>COMMENT</u>]	text

Explanation of Terms:

relation ::= {PRESENT number (, number)}
 {ABSENT number (, number) }

FUNCTION:

Attributes of the specified CLASSIFIER are ADDED. Attributes to be ADDED must be preceded by their labels.

RESTRICTIONS:

1. The CLASSIFIER must exist in the current VERSION.
2. Attributes can be ADDED only if they have not been previously ENTERed or generated by the system.
3. To ADD either TYPE or a PARENT/FOSTER selection, the relations between these two attributes must remain consistent. (See RESTRICTION 3 of ENTER CLASSIFIER.)
4. The newly ADDED attributes cannot duplicate any original defined attributes.

EXAMPLES:

1. To ADD to CLASSIFIER 2 the attribute of having a FOSTER parent of 56 when CLASSIFIER 7 is PRESENT:

SASE: ADD CLA 2 FOS 56 PRE 7

2. If the TYPE of CLASSIFIER 4 is TRA, but the user tries to ADD a single PARENT:

SASE: ADD CLA 4 PAR 8

?DO75 CONFLICT BETWEEN CLASSIFIER'S TYPE AND PARENT FOSTER.

ERRORS:

1. Attempt to access an undefined CLASSIFIER.
2. Attempt to ADD attributes previously defined.
3. Inconsistency between TYPE and PARENT/FOSTER designation.
4. CLASSIFIER, PARENT, FOSTER or CONTEXT relation numbers are not unique in the current VERSION.

EXAMPLES:

1. To DELETE the NAME, TITLE and COMMENT of CLASSIFIER 423.

SASE: DEL CLA 423 NA T COM

2. To DELETE ALL of the FOSTER parents of CLASSIFIER 1008:

SASE: DEL CLA 1008 FOS ALL

3. To DELETE the FOSTER parent 37 (of CLASSIFIER 401) plus ALL of its CONTEXT relations:

SASE: DEL CLA 401 FOS 37 CON ALL

4. To DELETE the FOSTER parent 116 (of CLASSIFIER 401) plus its CONTEXT relations 256 and 6000:

SASE: DEL CLA 401 FOS 116 CON 256, 6000

5. To DELETE the entire CLASSIFIER 201:

SASE: DEL CLA 201 ALL

CONFIRM TO DELETE THE ENTIRE (CLA) 201. [Y/N]? Y

** DELETION COMPLETED **

ERRORS:

1. Attempt to access an undefined CLASSIFIER.
2. Attempt to DELETE undefined attributes.
3. Attempt to DELETE undefined FOSTER parents.

DISPLAY CLASSIFIER

FORMAT:

```
DISPLAY CLASSIFIER [REFERENCE] | ALL | | | ALL | | | [PRINTER]  
| number | | | | | |  
| number-number | | | (labels) | | |
```

Explanation of Terms:

```
labels ::= { REFERENCE  
            NAME  
            TITLE  
            TYPE  
            PARENT  
            FOSTER  
            COMMENT }
```

FUNCTION:

DISPLAY CLASSIFIER ALL is used to DISPLAY ALL CLASSIFIERS in the current VERSION. DISPLAY CLASSIFIER number is used to DISPLAY attributes of a specific CLASSIFIER. DISPLAY CLASSIFIER number-number is used to DISPLAY all CLASSIFIERS from the first REFERENCE number to the second REFERENCE number, inclusive.

If the second ALL is used, ALL existing attributes of the CLASSIFIER(s) are DISPLAYed. Otherwise, only the attributes corresponding to the indicated labels are DISPLAYed.

RESTRICTIONS:

1. In DISPLAY CLASSIFIER ALL, at least one CLASSIFIER must exist in the current VERSION.
2. In DISPLAY CLASSIFIER number, the specified CLASSIFIER number must exist.
3. In DISPLAY CLASSIFIER number-number, at least one CLASSIFIER must exist between the numbers given.

EXAMPLES:

1. To DISPLAY ALL CLASSIFIERS between the REFERENCE numbers of 1101 and 1104 including TITLES and PARENTs:

SASE: DIS CLA 1101-1104 T PAR

REFER	PAREN	TITLE
1101		BUILDING

.....

REFER	PAREN	TITLE
1102	1101	PART OF BUILDING

.....

REFER	PAREN	TITLE
1103	1101	SAI BUILDING

.....

REFER	PAREN	TITLE
1104	1102	HIGH BUILDINGS

.....

2. To DISPLAY ALL existing attributes of CLASSIFIER 1101:

SASE: DIS CLA 1101 ALL

REFER	NAME	TYPE	PAREN	TITLE
1101	C1101	TRA		BUILDING

FOSTER: < 02 CONTEXT -06 04 >

COMMENT:

.....

* The minus sign in the CONTEXT REFERENCE number indicates ABSENT, while a positive number indicates a PRESENT CONTEXT relation. Consequently, in this example, the CLASSIFIER may have a FOSTER parent 2 if CLASSIFIER 6 is ABSENT and CLASSIFIER 4 is present.

ERRORS:

1. The CLASSIFIER is not defined.
2. The DISPLAY requested contains no CLASSIFIERS.

MODIFY CLASSIFIER

FORMAT:

MODIFY CLASSIFIER [REFERENCE] number

}	[<u>N</u> AME]	name	}
	[<u>T</u> ITLE]	text	
	[<u>T</u> YPE]	<u>A</u> CTUAL	
		<u>T</u> RANSSPARENT	
	[<u>P</u> ARENT]	number	
	[<u>F</u> OSTER]	{ number, number (; number , number) (number <u>C</u> ONTEXT number, number (; number, number)) }	
	[<u>C</u> OMMENT]	text	

FUNCTION:

Attributes of the specified CLASSIFIER are MODIFIED.

Any MODIFICATION of FOSTER parent(s) or their CONTEXT relation(s) must consist of a pair of REFERENCE numbers; old number, new number. The new number replaces the old (original) number.

RESTRICTIONS:

1. The CLASSIFIER must exist in the current VERSION.
2. All attributes previously defined can be modified, except TYPE.
3. If the CLASSIFIER has a defined PARENT or FOSTER parent(s), the default value of TYPE can not be MODIFIED.

EXAMPLES:

1. To MODIFY the NAME and PARENT of CLASSIFIER 1121:

SASE: MOD CLA 1121 NA OBJECT PARENT 2001

2. To MODIFY FOSTER parent 3000 to 4000 of CLASSIFIER 203:

SASE: MOD CLA 203 FOS 3000, 4000

3. To MODIFY the two CONTEXT relations of FOSTER parent 100 (of CLASSIFIER 789); namely, change 200 to 300 and 250 to 350:

SASE: MOD CLA 789 FOS 100 CON 200, 300; 250, 350

ERRORS:

1. The CLASSIFIER is not defined.
2. Attempt to MODIFY undefined attributes.
3. Conflict between the CLASSIFIER's TYPE and the PARENT/FOSTER designation.
4. Attempt to MODIFY undefined FOSTER parents.

GENERATE TREE

FORMAT:

GENERATE TREE [REFERENCE] number

[{	[<u>ORDER</u>]								}]
		[<u>ELSE</u>]									
		[<u>SORT</u>]									
		[<u>SAVE</u>]									

FUNCTION:

A decision TREE is GENERATED from a designated decision TABLE.

GENERATE TREE ORDER specifies the sequence in which CONDITION rows are selected for testing. INPUT specifies that CONDITIONS are to be tested in the original INPUT sequence. The QUICK and DELAY qualifiers operate as follows:

1. First Priority: Find the CONDITION row with the minimum number of immaterial entries (. or I).
2. Second Priority: Find the CONDITION row with the maximum number of explicit entries (T, Y, N or F).
3. Third Priority:
 - * For the QUICK qualifier: Find the CONDITION row with the maximum difference between the number of true (T or Y) and false (N or F) entries.
 - * For the DELAY qualifier: Find the CONDITION row with the minimum difference between the number of true (T or Y) and false (N or F) entries.

If an ORDER qualifier is not indicated, its default value is INPUT.

GENERATE TREE ELSE causes all ELSE RULEs to be GENERATED and ADDED to the defined RULEs in the TABLE.

GENERATE TREE SORT rearranges the TREE and the corresponding TABLE so that the shorter branches of the TREE are DISPLAYED before the longer branches.

The decision TABLE and decision TREE GENERATED by ORDER and/or SORT may be

SAVED for subsequent processing. The SAVED TREE can only be DELETED when DELETE TREE is used. The unSAVED GENERATED TREES are DELETED when DELETE TREE is used, when the user leaves the system, or when a new GENERATE TREE command is issued.

RESTRICTIONS:

1. The REFERENCE number of the GENERATED TREE must correspond to a defined TABLE which is complete, non-redundant, and non-contradictory (i.e., no two RULES can exist with identical CONDITION entries or vice-versa).
2. A TREE in excess of 120 nodes can not be GENERATED.

EXAMPLES:

1. To GENERATE a TREE corresponding to a defined TABLE 1124:

```
SASE: GEN TRE 1124
** TREE 1124 IS SUCCESSFULLY GENERATED.
```

2. To GENERATE a TREE ORDERed by the original INPUT sequence and SORTed:

```
SASE: GEN TRE 1234 ORD INP SOR
** TREE 1234 IS SUCCESSFULLY GENERATED
```

3. To GENERATE the ELSE RULEs in a defined TABLE 1000:

```
SASE: GEN TRE 1000 ELSE
** THE FOLLOWING ELSE RULE(S) ARE DETECTED:
```

```
RUL      1  2
```

```
-----
```

```
C1      T  F
C2      F  T
```

```
-----
```

```
** TREE 1000 IS SUCCESSFULLY GENERATED.
```

ERRORS:

1. The TREE REFERENCE number does not correspond to a defined TABLE REFERENCE number in the current CHAPTER.
2. The TREE has been previously GENERATED and SAVED.

DELETE TREE

FORMAT:

DELETE TREE [REFERENCE] number

FUNCTION:

The specified TREE is DELETED.

RESTRICTIONS:

1. The TREE must be previously GENERATED.

EXAMPLES:

1. To DELETE a defined TREE 1111:

SASE: DEL TRE 1111

** DELETION COMPLETED **

ERRORS:

1. The TREE does not exist in the current CHAPTER.

DISPLAY TREE

FORMAT:

DISPLAY TREE [REFERENCE] number [STUB] [COMMENT] [PRINTER]

FUNCTION:

The previously GENERATED TREE and its corresponding TABLE are DISPLAYed. If the ELSE qualifier was specified when the TREE was GENERATED, the TABLE for the ELSE RULEs are also DISPLAYed.

If the STUB qualifier is used, all existing STUBs of the CONDITION and /or ACTION entries are DISPLAYed. If the PRINTER qualifier is used, the output is sent to a PRINT file for off-line PRINTing. Otherwise, the DISPLAY is produced on-line.

RESTRICTIONS:

1. The TREE must be defined in the current CHAPTER.

EXAMPLES:

1. To DISPLAY a defined TREE and its CONDITION/ACTION STUBs:

```
SASE: DIS TRE 1236 STUB
```

```
DECISION TABLE  1  2
-----
1 EARTHQUAKE 1 *T  F
2 RESISTANCE      *T  +
*****
1 CANCEL 1        *X
2 PROCESS         *   X
-----
```

```
DECISION TREE
```

```
C1 + + C2 + + R1
-      -
-      - - ELSE
-
- - - R2
```

ERRORS:

1. The TREE does not exist in the current CHAPTER.

4. GENERATE NETWORK does not recognize disconnected single nodes (a DATUM that is not related to any other DATUM; i.e., it has no ingredients or dependents.)

EXAMPLES:

1. To GENERATE the NETWORK for CHAPTER 1 with LEVEL as the primary key SORTed in descending order and FLOAT as the secondary key SORTed in descending order:

SASE: GEN NET WIT CHA 1 SOR LEV SMA SMA SAV

NUMBER OF NODES AT LEVEL 0 FROM OUTPUT: 1

REF NUMBERS OF THESE NODES ARE: 1100

NUMBER OF NODES AT LEVEL 0 FROM INPUT: 3

REF NUMBERS OF THESE NODES ARE: 601 98 76

2. To GENERATE the NETWORK for ALL CHAPTERs in the current VERSION and store for further processing:

SASE: GEN NET WIT ALL SAVE

THE FOLLOWING LOOP IN THE NETWORK IS DETECTED:

11- 18- 201- 98-

ERRORS:

1. A NETWORK exists for the VERSION.
2. No data items defined in the current CHAPTER or VERSION.
3. INGREDIENTS are not related within the DATUM of the current CHAPTER or VERSION.
4. A loop is detected.

DELETE NETWORK

FORMAT:

DELETE NETWORK

FUNCTION:

The NETWORK of the current version is DELETED. Prior to the DELETion, the system will prompt the user for confirmation.

RESTRICTIONS:

1. The NETWORK must exist.

EXAMPLES:

1. To DELETE the NETWORK of the current VERSION:

SASE: DEL NET

** CONFIRM TO DELETE THE ENTIRE [NET]
OF THE CURRENT VERSION [Y/N] ? Y

** DELETION COMPLETED **

ERRORS:

1. NETWORK does not exist.

sub-network rooted at a specified node may be requested. If the word TITLE is specified, the TITLES of each data item are DISPLAYed; otherwise, only the DATUM REFERENCE numbers are shown.

If the LEVEL option is chosen, either the maximum or minimum number of input or output levels to be DISPLAYed may be specified.

The conventions used in the displayed output are:

1. A "-" before the DATUM number indicates that the node has previously occurred in the NETWORK.
2. A "*" following the DATUM number indicates that the node has INGREDIENTS (is a derived node) and the sub-NETWORK rooted at that node has previously occurred in the NETWORK.

DISPLAY NETWORK DATALIST provides a tabular DISPLAY analogous to DISPLAY DATUM ALL, except that the derived attributes DEPENDENTS, FLOAT and LEVEL may also be included in the listing.

RESTRICTIONS:

1. The NETWORK must exist.
2. To DISPLAY a segment of the NETWORK, the specified REFERENCE number must be an input or output node.

EXAMPLES:

1. To DISPLAY the ENTIRE DEPENDENCE NETWORK including TITLES:

```
SASE: DIS NET DEP ENT T
COMPLETE GLOBAL DEPENDENT NETWORK
EXTREME LEVEL FROM INPUT
0      1      2
...100 COMPONENT
      :.....71 MATERIAL
      :.....88 WOOD
      :.....90 STONE
```

2. To DISPLAY a tabular listing of the NETWORK including the attribute FLOAT:

SASE: DIS NET DAT FLO

110 FLOAT 0

160 FLOAT 1

170 FLOAT 0

180 FLOAT 0

ERRORS:

1. NETWORK does not exist.
2. Root node is not defined.

GENERATE HIERARCHY

FORMAT:

```
GENERATE HIERARCHY      || ALL ||      [SAVE]  
                          || FACET ||
```

FUNCTION:

The HIERARCHY for the current VERSION is GENERATED from the existing CLASSIFIERS.

If GENERATE HIERARCHY FACET is used, the CLASSIFIER HIERARCHY is GENERATED through the PARENTs only, terminating at the root nodes of FACETs, which are the CLASSIFIERS having no PARENT.

If GENERATE HIERARCHY ALL is used, the FACETs are combined into several FIELDS by first connecting the root nodes of the FACETs and then proceeding upward through the FOSTER parents to the root nodes of the FIELDS. The inclusion of a FACET root node into the FIELD is governed as follows:

- * If a node has only one FOSTER parent, the node is automatically entered into the current constructed FIELD HIERARCHY stack.
- * If a node has more than one FOSTER parent, the node is conditionally entered; depending on whether the specified CONTEXT relations are PRESENT or ABSENT in the current constructed FIELD HIERARCHY stack.

If no errors are detected, and the SAVE qualifier is specified, the resulting HIERARCHY is stored and is available for further processing.

RESTRICTIONS:

1. Only one HIERARCHY can exist in the current VERSION.
2. No HIERARCHY can be GENERATED if:
 - a: The VERSION contains no CLASSIFIERS.
 - b: The CLASSIFIERS contained in the VERSION constitute a loop. (In this case, the list of CLASSIFIERS composing the loop is DISPLAYed.)
3. A detached CLASSIFIER, one which has no PARENT or FOSTER parent, represents a complete FACET. These isolated CLASSIFIERS prevent the execution of GENERATE HIERARCHY ALL. However, the detached CLASSIFIERS are DISPLAYed on the listing in this case.

4. To GENERATE HIERARCHY ALL, at least one FOSTER parent relation must exist among the CLASSIFIERS so that FIELD can be GENERATED.

EXAMPLES:

1. To GENERATE the HIERARCHY through the PARENTs only:

SASE: GEN HIE FAC SAVE

1230 1210 1220

?D094 A HIERARCHY LOOP ABOVE IS DETECTED.

2. Attempt to GENERATE a FIELD HIERARCHY:

SASE: GEN HIE ALL COM 'FOR TESTING'

10 77 124

?D095 CLASSIFIERS SHOWN ABOVE ARE ISOLATED/DETACHED

3. If no FOSTER parent relations are defined in the version, HIERARCHY can not be GENERATED through FIELD.

SASE: GEN HIE ALL SAVE

?D097 NO SINGLE FIELD EXISTS, ALL FACETS.

ERRORS:

1. The HIERARCHY has been previously GENERATED or SAVED.
2. There are no defined CLASSIFIER(s) in the current VERSION.
3. Detached CLASSIFIERS prevent the GENERATION of HIERARCHY ALL.
4. No single FIELD exists.
5. A HIERARCHY loop is detected.

DELETE HIERARCHY

FORMAT:

DELETE HIERARCHY

FUNCTION:

The entire HIERARCHY is DELETED. Prior to the DELETion, the system will prompt the user for confirmation.

RESTRICTIONS:

1. The HIERARCHY must exist in the current VERSION.

EXAMPLES:

1. To DELETE the entire HIERARCHY:

```
SASE: DEL HIE
** CONFIRM TO DELETE THE ENTIRE [HIE] OF THE
   CURRENT VERSION [Y/N]? Y
** DELETION COMPLETED **
```

ERRORS:

1. The HIERARCHY does not exist in the current VERSION.

DISPLAY HIERARCHY

FORMAT:

```

DISPLAY HIERARCHY || ALL ||
                    ||
                    || ROOTS || FIELD || || STRUCTURE [TITLE] ||
                    ||         || FACET || ||
                    || FIELD || number || || LIST (classifier ||
                    || FACET ||         || || labels) ||
                    ||
                    || COUNT || || PRINTER ||

```

Explanation of Terms:

classifier labels ::= { REFERENCE
NAME
TITLE
TYPE
PARENT
FOSTER }

FUNCTION:

DISPLAY HIERARCHY ALL is used to DISPLAY the entire list of CLASSIFIERS composing the HIERARCHY of the current VERSION. The DISPLAY consists of both FIELDS and FACETS presented in ascending order according to the REFERENCE numbers of the ROOT nodes. Each FIELD and FACET is preceded by a heading FIELD n or FACET n, where n is an ordinal number 1, 2, etc.

If DISPLAY HIERARCHY ROOTS is used, only the designated FIELD or FACET ROOTs are DISPLAYed.

If DISPLAY HIERARCHY FIELD number or FACET number is used, only the portion of the HIERARCHY ROOTed at that specified CLASSIFIER is DISPLAYed.

The STRUCTURE qualifier generates a structured output (with variable indentation) including optional TITLES. The LIST qualifier generates a tabular output with optional CLASSIFIER labels.

If COUNT is requested, the total number of CLASSIFIERS within each FIELD or FACET is also DISPLAYed.

If PRINTER is requested, the output is stored in a file for off-line PRINTing. Otherwise, the output is displayed.

RESTRICTIONS:

1. The HIERARCHY must be defined in the current VERSION.
2. If only FACETs are GENERATED in the HIERARCHY, neither DISPLAY HIERARCHY ALL or DISPLAY HIERARCHY ROOTS FIELD can be specified.
3. To DISPLAY a specific FIELD or FACET, the CLASSIFIER REFERENCE number must be a ROOT node of the FIELD or FACET, respectively.

EXAMPLES:

1. To DISPLAY a FACET HIERARCHY with a ROOT node of 100 in a structured output, including a COUNT of the total number of CLASSIFIERS:

SASE: DIS HIE FAC 100 STR COU

FACET 18 COUNT: 6

100

101

102

103

104

105

2. To DISPLAY both FACETs and FIELDS in a tabular output with TITLES, NAMES, and PARENTs:

SASE: DIS HIE ALL LIS T NA PAR

REFER	NAME	PAREN	TITLE
4	C14		

FACET: 1

FIELD: 1

.....

REFER	NAME	PAREN	TITLE
2	C12	4	

.....

ERRORS:

1. The HIERARCHY does not exist in the current VERSION.
2. Attempt to DISPLAY a FIELD when only FACET(s) have been GENERATED in the HIERARCHY.
3. The FIELD/FACET number specified is not a ROOT node.

GENERATE SCOPELIST

FORMAT:

```
GENERATE SCOPELIST WITH [CHAPTER]      || ALL ||  
                                         || name (,name) ||  
  
[SAVE]
```

FUNCTION:

A SCOPELIST is GENERATED for each CLASSIFIER in the HIERARCHY of the current VERSION. A SCOPELIST is the list of all DATUMs having this CLASSIFIER in its ARGUMENT attribute list. If GENERATE SCOPELIST WITH CHAPTER ALL is used, the DATUMs are taken from ALL of the CHAPTERs in the current VERSION. If GENERATE SCOPELIST WITH CHAPTER name (,name) is used, the DATUMs are taken from the specified CHAPTERs.

IF the SAVE qualifier is used, the GENERATED SCOPELIST is stored and is available for further processing. It can be DELETED only when DELETE SCOPELIST is used.

RESTRICTIONS:

1. Either HIERARCHY ALL or HIERARCHY FACET must have been GENERATED in the current VERSION.
2. At least one DATUM must exist in the selected CHAPTER(s), with some CLASSIFIERs defined as ARGUMENTs.
3. Only one SCOPELIST can exist in any one VERSION.

EXAMPLES:

1. To GENERATE a SCOPELIST with DATUMs taken from ALL available CHAPTERs:
SASE: GEN SCO WIT CHA ALL
2. To GENERATE a SCOPELIST with DATUMs taken only from CHAPTERs 1, 2 and 3:
SASE: GEN SCO WIT CHA ONE, TWO, THREE

ERRORS:

1. A SCOPELIST has been previously GENERATED.

2. No CHAPTER is defined in the current VERSION or the indicated CHAPTER(s) does not exist.
3. No DATUM is contained in the specified CHAPTER(s).
4. No ARGUMENTs are defined in the DATUM(s).

DELETE SCOPELIST

FORMAT:

DELETE SCOPELIST

FUNCTION:

The current SCOPELIST is DELETED. Prior to DELETion, the system will prompt for confirmation.

RESTRICTIONS:

1. To DELETE the SCOPELIST:

```
SASE: DEL SCO
** CONFIRM TO DELETE THE ENTIRE [SCO] OF THE
   CURRENT VERSION. [Y/N]? Y
** DELETION COMPLETED **
```

ERRORS:

1. The SCOPELIST does not exist in the current VERSION.

DISPLAY SCOPELIST

FORMAT:

DISPLAY SCOPELIST [REFERENCE] number

<u>ALL</u>		<u>STRUCTURE</u>		
number		<u>ALL</u>		
number-number		<u>LIST</u> (classifier labels)		
<u>FIELD</u> number(, number)				
<u>FACET</u>				
<u>COUNT</u>				
<u>ALL</u>	<u>ALL</u>	<u>ALL*</u>		
<u>REQUIREMENTS</u>	<u>INDEX</u>	(datum labels)		[<u>PRINTER</u>]
<u>DETERMINATION</u>	<u>OUTLINE</u>			

Explanation of Terms:

datumlables ::= {
REFERENCE
NAME
TITLE
SECTION
PAGE
VALVE
SOURCE
TYPE
STATUS
UTILIZATION
INGREDIENTS
ARGUMENTS
EQUIVALENTS
}

*Note: If STRUCTURE is used, then TITLE is the only datum label that can be displayed.

classifierlabels ::= {
REFERENCE
NAME
TITLE
PARENT
FOSTER
}

FUNCTION:

The current SCOPELIST is DISPLAYed in ascending order according to the CLASSIFIER REFERENCE numbers.

DISPLAY SCOPELIST ALL is used to DISPLAY the SCOPELIST(s) of ALL pertinent CLASSIFIERS. DISPLAY SCOPELIST number is used to DISPLAY the SCOPELIST of a specified CLASSIFIER. DISPLAY SCOPELIST number-number is used to DISPLAY the SCOPELIST(s) of all CLASSIFIERS from the first REFERENCE number to the second REFERENCE number, inclusive. DISPLAY FIELD or FACET number(,number) is used to DISPLAY the SCOPELIST of all CLASSIFIERS within the FIELD(s) or FACET(s) rooted at the specified CLASSIFIER REFERENCE number(s).

If the STRUCTURE qualifier is used, CLASSIFIERS and their corresponding provisions are printed out in a structured form (with variable indentation). TITLE is the only DATUM label that can be DISPLAYed. The LIST qualifier generates a tabular output. If LIST ALL is used, ALL existing attributes of each CLASSIFIER are DISPLAYed. Otherwise, only the attributes corresponding to the indicated labels are DISPLAYed.

The output can consist of either the COUNT of the total number of DATUMS in the SCOPELIST of each CLASSIFIER, or pertinent information regarding the DATUMS. More specifically, the SCOPELIST of each CLASSIFIER can include:

1. ALL DATUMS having this CLASSIFIER as an ARGUMENT, or
2. those DATUMS labeled as REQUIREMENTS, or
3. those DATUMS labeled as DETERMINATIONS.

The SCOPELIST can also include:

1. ALL possible DATUMS, or
2. those DATUMS which possess ARGUMENTS defined to be used for INDEXing only, or
3. those DATUMS which possess ARGUMENTS defined to be used for OUTLINE.

For each DATUM, ALL defined attributes can be DISPLAYed if specified. Otherwise, only the attributes corresponding to the indicated DATUM labels are included.

RESTRICTIONS:

1. A SCOPELIST must exist.
2. In DISPLAY SCOPELIST ALL, at least one CLASSIFIER REFERENCE number must be defined in the current VERSION.
3. In DISPLAY SCOPELIST number, the specified CLASSIFIER REFERENCE number must be defined.

4. In DISPLAY SCOPELIST number-number, at least one CLASSIFIER REFERENCE number must be defined between the numbers given.
5. In order to DISPLAY the SCOPELIST of all CLASSIFIERS within a FACET or FIELD, HIERARCHY must first be GENERATED. Furthermore, if the HIERARCHY was GENERATED through FACETS only, the SCOPELIST of CLASSIFIERS within a FIELD can not be DISPLAYed.
6. The CLASSIFIER number corresponding to either FIELD or FACET must represent a root node.

EXAMPLES:

1. Tabular output of CLASSIFIERS 1101-1102:

SASE: DIS SCO 1101-1102 LIS R ALL ALL ALL

REFER
1101

REFER	NAME	SECTI	PAGE	VALUE	SOURC	TYPE
11					INPUT	FUNTN

TITLE: BUILDING
 INGREDIENTS:
 ARGUMENTS: 07 04
 ARG FOR OUL: 1101 1102 06
 EQUIVALENTS:

.....

REFER
1102

REFER	NAME	SECTI	PAGE	VALUE	SOURC	TYPE
11					INPUT	FUNTN

TITLE: BUILDING
 INGREDIENTS:
 ARGUMENTS: 07 04
 ARG FOR OUL: 1101 1102 06
 EQUIVALENTS:

.....

2. STRUCTURED output of CLASSIFIERS 1101-1104

SASE: DIS SCO 1101-1104 STR ALL ALL ALL

CLASSIFIER	PROVISION
-----	-----
1001 CATEGORY A	11 TITLE: BUILDING
1102 CATEGORY B	11 TITLE: BUILDING
1103 BUILDING MATERIAL	12 TITLE:
1104 PART OF BUILDING	12 TITLE:

ERRORS:

1. SCOPELIST is not defined in the current VERSION.
2. HIERARCHY of FIELD is not defined:
3. CLASSIFIERS are not defined.
4. The DISPLAY requested contains no CLASSIFIERS.
5. Attempt to DISPLAY undefined attributes.

GENERATE INDEX

FORMAT:

```
GENERATE INDEX    || SIMPLE                ||
                  || SUBDIVIDED number1  ||
                  ||
                  || [ DROP { CLASSIFIER number2 (,number2) } ]
                  || { ROOTS number2 (,number2) }
                  || SCOPE || number1 ||
                  || number1 PERCENT ||
```

Explanation of Terms

Number1 - Ordinal number
Number 2 - Classifier reference number

FUNCTION:

An index for the SCOPELIST of the current VERSION is GENERATED. It is an alphabetical listing of the TITLES of all existing CLASSIFIERS. The SCOPELIST (all DATUMs having this CLASSIFIER as a ARGUMENT) of each CLASSIFIER is also GENERATED.

GENERATE INDEX SIMPLE is used to GENERATE one level of headings. GENERATE INDEX SUBDIVIDED number1 is used for the GENERATION of a hierarchical INDEX. Such an INDEX is SUBDIVIDED according to CLASSIFIER level, so as to break up large clusters of provisions.

The DROP qualifier can eliminate the following CLASSIFIERS from the INDEX:

1. All CLASSIFIERS specified by their REFERENCE numbers; or
2. All CLASSIFIERS within the FIELD(s) or FACET(s) ROOTed at the specified CLASSIFIER REFERENCE number(s); or
3. Those CLASSIFIERS whose scope exceeds either a given number1 or a percentage of the total number of provisions in the current VERSION.

RESTRICTIONS:

1. INDEX can be GENERATED only if a SCOPELIST has been GENERATED in the current VERSION.
2. INDEX can be GENERATED only if a HIERARCHY has been GENERATED in the current VERSION.
3. The CLASSIFIERS to be DROPPed must be defined in the current VERSION.
4. No FIELD or FACET can be DROPPed if the indicated ROOT number has no INGREDIENTS.

EXAMPLES:

1. To GENERATE an INDEX with 1 level of headings excluding CLASSIFIERS 20 and 30:

SASE: GEN IND SIM DRO CLA 20, 30

2. To GENERATE a hierarchical INDEX:

SASE: GEN IND SUB 4 DRO SCO 5 P

ERRORS:

1. SCOPELIST does not exist in the current VERSION.
2. An INDEX exists for the current VERSION.
3. Specified CLASSIFIER is not defined in the current VERSION.
4. Specified CLASSIFIER is not a ROOT of a FIELD or FACET.

DELETE INDEX

FORMAT:

DELETE INDEX

FUNCTION:

The INDEX of the current VERSION is DELETED. Prior to DELETion, the system will prompt the user for confirmation.

RESTRICTIONS:

1. The INDEX must exist in the current VERSION.

EXAMPLES:

1. To DELETE the current INDEX:

SASE: DEL IND

** CONFIRM TO DELETE THE ENTIRE [IND] OF THE
CURRENT VERSION. [Y/N] ? Y

** DELETION COMPLETED **

ERRORS:

1. The INDEX does not exist in the current VERSION.

4. All DATUMs labeled as DETERMINATIONs for each CLASSIFIER.

For each DATUM, ALL existing attributes can be DISPLAYed if specified. Otherwise, only the attributes corresponding to the indicated DATUM labels are DISPLAYed.

RESTRICTIONS:

1. INDEX must exist
2. The CLASSIFIER number(s) corresponding to either FIELD or FACET must be a root node.
3. The attributes to be DISPLAYED must be defined.

EXAMPLES:

1. To DISPLAY the INDEX of ALL existing CLASSIFIERs plus ALL of their provisions and provisional attributes:

SASE: DIS IND ALL ALL ALL

ALPHABETICAL INDEX

(NO TITLE FOR CLASSIFIER NUMBER 1103)

(NO TITLE FOR CLASSIFIER NUMBER 1104)

(NO TITLE FOR CLASSIFIER NUMBER 1112)

REFER	NAME	SECTI	PAGE	VALUE	SOURCE	TYPE	STATU	UTILI
12		TWO	234		INPUT		CLASS	

TITLE: THIS IS A TITLE FOR DATUM #12

INGREDIENTS:

ARG FOR IND: 1112 113 115 4

ARG FOR OUT: 1103 1104

EQUIVALENTS:

- * The line with "*" is used to separate each CLASSIFIER and the dotted line is used to separate each DATUM within one CLASSIFIER. If no TITLE for a CLASSIFIER exists, the message "(NO TITLE FOR CLASSIFIER number)" is printed.
- 2. To DISPLAY the INDEX of ALL of the CLASSIFIERS plus a COUNT of the number of provisions of each CLASSIFIER:

SASE: DIS IND ALL COU

ALPHABETICAL INDEX

BUILDING	COUNT: 1
CATEGORY A	COUNT: 1
CATEGORY B	COUNT: 1
MATERIAL NATURE OF BLDG.	COUNT: 1
PART OF BUILDING	COUNT: 1
SEISMIC PERFORMANCE	COUNT: 1
SPECIFIC BUILDING	COUNT: 1

ERRORS:

1. The INDEX does not exist in the current VERSION.
2. Specified CLASSIFIER(s) are not defined.
3. Specified CLASSIFIER(s) fail to be the root node(s) of FIELD or FACET.
4. Attempt to DISPLAY undefined attributes.

ENTER ORGANIZATION

FORMAT:

ENTER ORGANIZATION [REFERENCE] number

{ || LEVEL || (labels) }

{ || PARENT || }

Syntax of Dialogue:

User entry: || number1 [,number2] ||

|| END ||

Explanation of Terms:

number1 - classifier reference number

number2 - level number of heading number

labels ::= { NAME }

{ TITLE }

{ COMMENT }

FUNCTION:

An ORGANIZATIONAL network is ENTERed under the current VERSION. The ORGANIZATIONAL network is a hierarchy ordered by heading numbers and indented by levels. Each heading is a single CLASSIFIER. (The SCOPELIST of each CLASSIFIER is not included in the ORGANIZATIONAL network.)

ENTER ORGANIZATION number LEVEL is used to ENTER an ORGANIZATIONAL network by assigning specific levels to individual CLASSIFIERS. ENTER ORGANIZATION number PARENT is used to ENTER an ORGANIZATIONAL network with the entry of CLASSIFIERS governed by the PARENT of individual CLASSIFIERS.

The ORGANIZATIONAL network is created a heading at a time. After the initial input line, a dialogue begins between the user and SASE. A system prompt of "D:" signals the user to begin ENTERing input; consisting of:

number1 [,number2]

where number1 is the REFERENCE number of the CLASSIFIER designated to serve as the heading. Number2 is either the LEVEL number assigned to the current CLASSIFIER. If number2 is not ENTERed, the default value is 1 for the first heading. Otherwise, if number2 is not ENTERed, the system assumes the value of number2 to be the previous LEVEL or PARENT number.

After each entry, the system responds with the sequential heading number, the LEVEL number and the TITLE of the current CLASSIFIER. The dialogue is terminated by the user by ENTERing the qualifier END.

RESTRICTIONS:

1. The CLASSIFIERS designated as headings must be defined in the current VERSION.
2. Once a CLASSIFIER is ENTERed into the ORGANIZATION, the DELETion of that CLASSIFIER does not affect the existing ORGANIZATION. However, a warning message is provided during the execution of DISPLAY ORGANIZATION.
3. The LEVEL or PARENT numbers need not be sequential.

EXAMPLES:

1. To ENTER an ORGANIZATION by assigning LEVELs to the CLASSIFIERS:

```
SASE: ENT ORG 1 LEV
D: 200
HED 1 LEV 1 : BUILDING
D: 300, 2
HED 2 LEV 2 : SPECIFIC BUILDING
D: 400
HED 3 LEV 2 : PART OF BUILDING
D: END
SASE:
```

2. To ENTER an ORGANIZATION by specifying PARENT(s) of CLASSIFIERS for each heading:

```
SASE: ENT ORG 1 PAR
D: 200
HED 1 LEV 1 : BUILDING
D: 300, 1
HED 2 LEV 2 : SPECIFIC BUILDING
D: 400, 1
HED 3 LEV 2 : PART OF BUILDING
D: END
SASE
```

ERRORS:

1. ORGANIZATION was previously ENTERed.
2. The specified CLASSIFIER number is not defined in the current VERSION.

ADD ORGANIZATION

FORMAT:

ADD ORGANIZATION [REFERENCE] number

{	[<u>NAME</u>]	text	}
	[<u>TITLE</u>]	text	
	[<u>COMMENT</u>]	text	}

FUNCTION:

Attributes of the referenced ORGANIZATION are ADDED

RESTRICTIONS:

1. The referenced ORGANIZATION must be defined.
2. An attribute may be ADDED only if it has not been entered before.
3. Each attribute value must be preceded by its label.

EXAMPLES:

1. To ADD a TITLE for the referenced ORGANIZATION:
SASE: ADD ORG 1 T 'SEISMIC ORGANIZATION'
2. To ADD a COMMENT for the referenced ORGANIZATION:
SASE: ADD ORG 1 COM 'EDITION OF MAY, 1980'

ERRORS:

1. ORGANIZATION is not defined.
2. Attempt to ADD attributes previously defined.

CONTINUE ORGANIZATION

FORMAT:

CONTINUE ORGANIZATION [REFERENCE] number

Syntax of Dialogue:

```
User's entry :      || number1 [, number2] ||  
                  ||   END                ||
```

Explanation of Terms:

number1 - classifier reference number

number2 - level number of heading number

FUNCTION:

CONTINUE ORGANIZATION LEVEL/PARENT simply CONTINUES the ORGANIZATION dialogue (See ENTER ORGANIZATION). The user may choose either LEVEL or PARENT. To assist the user, SASE first prints out the last entry of the ORGANIZATIONAL network. Following this, a system prompt of "D:" appears signaling the user to ENTER any additional input.

RESTRICTIONS:

1. The ORGANIZATION must exist in the current VERSION.
2. See the RESTRICTIONS of ENTER ORGANIZATION.

EXAMPLES:

1. To ADD to the current ORGANIZATION:

```
SASE:   CON  ORG  PAR  
  
LAST ENTRY WAS  
HED 9  LEV 10  : MATERIAL  
D: 111, 9  
HED 10 LEV 11  : WOOD  
D: 121, 9  
HED 11 LEV 11  : STONE  
D:  END  
SASE:
```

ERRORS:

1. Attempt to access an undefined ORGANIZATION.

DELETE ORGANIZATION

FORMAT:

DELETE ORGANIZATION [REFERENCE] number

|| ALL ||
|| (labels) ||

Explanation of Terms:

labels ::= { NAME
 TITLE
 COMMENT }

FUNCTION:

If DELETE ORGANIZATION number ALL is used, the entire organization is DELETED. Otherwise, only the attributes corresponding to the indicated labels are DELETED.

RESTRICTIONS:

1. The ORGANIZATION must exist in the current VERSION.

EXAMPLES:

1. To DELETE the current ORGANIZATION:

SASE: DEL ORG 1 ALL

** CONFIRM TO DELETE THE ENTIRE [ORG] OF THE CURRENT
VERSION. [Y/N] ? Y

** DELETION COMPLETED **

ERRORS:

1. Attempt to DELETE an undefined ORGANIZATION.

DISPLAY ORGANIZATION

FORMAT:

DISPLAY ORGANIZATION [REFERENCE] number

```
|| STRUCTURE [TITLE] || [PRINTER]  
|| LIST (label2) ||  
|| (labels) ||
```

Explanation of Terms:

label2 - classifier labels
labels - see DELETE ORGANIZATION

```
label2 ::= { REFERENCE }  
           { NAME }  
           { TITLE }  
           { TYPE }  
           { PARENT }  
           { FOSTER }
```

FUNCTION:

DISPLAY ORGANIZATION is used to DISPLAY the ORGANIZATIOnal network. If the qualifier STRUCTURE is used, the ORGANIZATION is DISPLAYEed in a STRUCTUREd form, indented at each level. Each heading is followed by the CLASSIFIER REFERENCE number. The TITLE of each CLASSIFIER can also be DISPLAYed, if specified.

If the LIST qualifier is used, an unstructured ORGANIZATION LISTing is provided. ANY defined attribute of each CLASSIFIER REFERENCE number is followed by "*", it means that the CLASSIFIER has been DELETED from the current VERSION.

RESTRICTIONS:

1. An ORGANIZATION must exist in the current VERSION.

EXAMPLES:

1. To DISPLAY the current ORGANIZATION in a STRUCTUREd form including TITLES:

```
SASE: DIS ORG 1 STR T  
1 200 BUILDING  
2 300 SPECIFIC BUILDING  
3 400 PART OF BUILDING  
4 500 FUNCTION
```

2. To DISPLAY the current ORGANIZATION in a tabular form including TITLES:

SASE: DIS ORG 1 LIS T

HEAD	REFER	TITLE
1	200	BUILDING
2	300	SPECIFIC BUILDING
3	400	PART OF BUILDING
4	500	FUNCTION

ERRORS:

1. ORGANIZATION is not defined.
2. Attempt to DISPLAY undefined attributes.

MODIFY ORGANIZATION

FORMAT:

MODIFY ORGANIZATION [REFERENCE] number

}			<u>INSERT</u> [<u>HEADING</u>] number4			<u>PARENT</u>					
						<u>LEVEL</u>					
			<u>DELETE</u> [<u>HEADING</u>]			number4					
						number4 - number4					
			<u>DUPLICATE</u>			[<u>HEADING</u>] number4			<u>TO</u> number1		
			<u>MOVE</u>			number4 - number4					
<u>HEADING</u>	number4		[<u>REFERENCE</u>]number1			<u>LEVEL</u>	number3				
			<u>PARENT</u>			number4					
(labels)											

Syntax of Dialogue:

User's entry : | number1 [, number2] |
| END |

Explanation of Terms:

labels - see DELETE ORGANIZATION
number1 - classifier reference number
number2 - level or heading number
number3 - level number
number4 - heading number

FUNCTION:

Attributes of the current ORGANIZATION are MODIFIED.

MODIFY ORGANIZATION INSERT number4 PARENT/LEVEL is used to INSERT additional headings into the ORGANIZATIONAL network. The user may choose either PARENT or LEVEL. After the initial heading number is entered, a dialogue (similar to that in CONTINUE ORGANIZATION) begins between the user and SASE. The dialogue is terminated by the user when the qualifier END is entered. Upon termination, all existing heading numbers are resequenced.

MODIFY ORGANIZATION DELETE number4 is used to DELETE a specific heading. MODIFY ORGANIZATION DELETE number4 - number4 is used to DELETE all headings from the first heading number to the second heading number, inclusive. In either case, all heading numbers are resequenced accordingly.

MODIFY ORGANIZATION DUPLICATE/MOVE number4 TO number1 is used to DUPLICATE or MOVE a specific heading to a location indicated by a CLASSIFIER REFERENCE number. MODIFY ORGANIZATION DUPLICATE/MOVE number4 - number4 TO number1 is used to DUPLICATE or MOVE all headings from the first heading number to the second heading number, inclusive, to a location indicated by a CLASSIFIER REFERENCE number. In either case, all heading numbers are resequenced accordingly.

MODIFY ORGANIZATION HEADING number4 number1 is used to change the CLASSIFIER REFERENCE number at the specified heading. MODIFY ORGANIZATION HEADING number4 is used to change the assigned PARENT to the CLASSIFIER at the specified heading. All other headings remain unaffected. In each of these three cases, the heading numbers are not resequenced.

RESTRICTIONS:

1. The ORGANIZATION must be defined in the current VERSION.
2. All specified heading numbers or CLASSIFIER REFERENCE numbers must be defined in the current VERSION.
3. When MOVING several headings, the specified destination (CLASSIFIER REFERENCE number) cannot be included within the group of heading numbers.

EXAMPLES:

1. To change the LEVEL of heading number 18:

```
SASE: MOD  ORG  1  H  18  LEV  10
```

2. To DELETE all headings from number 36 to number 39, inclusive:

```
SASE: MOD  ORG  1  DEL  36-39
```

3. To begin the INSERTion process at heading number 21 including specified LEVELs.

SASE: MOD ORG 1 INS 21 LEV

HED 21 LEV 3 : MATERIAL

D: 111

HED 22 LEV 3 : BUILDING

D: 215, 5

HED 23 LEV 5 : STRUCTURE

:

:

D: END

SASE:

ERRORS:

1. ORGANIZATION is not defined.
2. Attempt to access an undefined heading number.
3. Attempt to access an undefined CLASSIFIER REFERENCE number.
4. Attempt to MOVE a group of headings to location (CLASSIFIER REFERENCE number) which is included in the group.

GENERATE OUTLINE

FORMAT:

GENERATE OUTLINE

```
|| FROM ORGANIZATION number || [ALL] USE || ALL ||  
|| STEPWISE || || REQUIREMENT ||  
|| || || DETERMINATION ||
```

```
[ DROP { CLASSIFIER number1 (, number1) }  
      { FIELD number1 (, number1) }  
      { FACET number1 (, number1) } ]
```

```
ENTRY || RELEVANT ||  
      || LOGICAL ||  
      || GRADED ||  
      || STRICT ||
```

Syntax of Dialogue:

```
User entry 1: || number1 [, number2] ||  
              || PARENT ||  
              || CLASSIFIER ||  
              || END ||
```

```
User entry 2: || number 3 (, number 3) ||  
              || (Null) ||
```

```
User entry 3: LIST
```

Explanation of Terms:

- number1 - classifier reference number
- number2 - level number
- number3 - provision (datum) reference number
- (Null) - blank line or carriage return
- LIST - To list potential datum associated with the classifier.

FUNCTION:

An OUTLINE for the current VERSION is GENERATED.

GENERATE OUTLINE FROM ORGANIZATION is used to GENERATE the OUTLINE directly FROM a previously ENTERed ORGANIZATION of the current VERSION. GENERATE OUTLINE STEPWISE is used to GENERATE the OUTLINE STEPWISE by direct user interaction with SASE. The user enters one heading at a time, in a manner similar to that of ENTER ORGANIZATION.

If the qualifier ALL is used, ALL headings defined in ORGANIZATION or STEPWISE are included in the OUTLINE. Otherwise, only those headings (CLASSIFIERS) with a defined potential SCOPELIST are included in the OUTLINE, where potential SCOPELIST is the list of DATUMs possessing the heading CLASSIFIER as an OUTLINING ARGUMENT and the same value of UTILIZATION as specified by the USE qualifier. In other words, the USE qualifier determines which DATUMs in the SCOPELIST of a CLASSIFIER are to be included in the OUTLINE. If the second ALL qualifier is used, ALL DATUMs are included in the OUTLINE. If the REQUIREMENT qualifier is used, only those DATUMs with USE = REQUIREMENT are included. If the DETERMINATION qualifier is used, only those DATUMs with USE = DETERMINATION are included in the OUTLINE.

The DROP qualifier can exclude the following CLASSIFIERS from the OUTLINE:

1. All CLASSIFIER(s) specified by their REFERENCE number(s); or
2. All CLASSIFIER(s) within the FIELD(s) rooted at the specified CLASSIFIER REFERENCE number(s); or
3. All CLASSIFIER(s) within the FACET(s) rooted at the specified CLASSIFIER REFERENCE number(s).

(DROP can not be used if the STEPWISE method is used to GENERATE the OUTLINE).

The ENTRY qualifier sets the criteria for matching a candidate DATUM with a heading. The criteria are based on the following objectives:

1. RELEVANT: Each CLASSIFIER on the branch must be related to one of the DATUMs OUTLINING ARGUMENTs.
2. COMPLETE: Each OUTLINING ARGUMENT of the DATUM must be included among the set of CLASSIFIERS on the branch.
3. UNIQUE: No CLASSIFIER on the branch may be an extended brother (brother, the brother of predecessors, and any descendant of such brothers) of any OUTLINING ARGUMENT of the DATUM.
4. GRADED: No CLASSIFIER on the branch may be a descendant of any OUTLINING ARGUMENT of the DATUM.

Based on these criteria, the possible choices of ENTERing a DATUM in the OUTLINE are defined as follows:

1. RELEVANT: Must satisfy the above Objective 1; that the logic of the FACET may be ignored. The user is allowed to ENTER DATUMs by their RELEVANCE. (This criterion applies only to the STEPWISE option.)
2. LOGICAL: Must satisfy the above Objectives 2 and 3.
3. GRADED: Must satisfy the above Objectives 2, 3 and 4.
4. STRICT: Must satisfy the above Objectives 1, 2 and 3.

In the STEPWISE option, a dialogue occurs between the user and SASE. A system prompt of "DC:" is used to signal the user to ENTER input. The user is asked to create a heading or group of headings each time by ENTERing:

```
number1 [, number2]
PARENT
CLASSIFIER
LIST
```

Where number1 is either the CLASSIFIER REFERENCE number chosen to be the next heading or the PARENT node for the set of CLASSIFIERS chosen to be the next set of headings. After the PARENT or CLASSIFIER is ENTERed, all subsequent number1(s) must represent either a PARENT node or a CLASSIFIER number, respectively; that is, until the reverse qualifier is specified. Number2 represents the level number. If number2 is not ENTERed, the default value is 1 for the first heading. Otherwise, if number2 is not ENTERed, the system assumes the value of number2 to be the previous level number.

If LIST is used, the system LISTs the potential DATUMs of the CLASSIFIER REFERENCE number ENTERed.

If number1 is ENTERed correctly, SASE identifies the potential DATUMs in the SCOPELIST of the heading CLASSIFIER, according to the ENTRY qualifier.

- * If the ENTRY is RELEVANT, the system prints out all of the potential provisions as follows:

```
RELEVANT DATUM: number3 ---
("NONE" if no potential datums)
```

- * If the ENTRY is not RELEVANT, the system output is shown as:

```
DATUM ALONG WITH THEIR UNMATCHED ARGUMENTS:
number3 ARGUMENTS LEFT: number1 ---
("NONE" if no unmatched argument)
```

In addition, the system supplies another prompt of "DD:" expecting the user to select the provisions to be included in the OUTLINE. The user can either ENTER a series of number3(s) separated by "," or press the carriage

return key if no provision is to be included.

The above cycle is repeated until the dialogue is terminated by ENTERing END.

RESTRICTIONS:

1. A SCOPELIST must be active in the current VERSION.
2. To GENERATE the OUTLINE from an ORGANIZATION, an ORGANIZATION, must be defined in the current VERSION.
3. To DROP a FIELD(s) or FACET(s), a current HIERARCHY must be active. The corresponding number1(s) must represent a FIELD or FACET root node.
4. All of the specified CLASSIFIER numbers must be defined in the current VERSION.

EXAMPLES:

1. To GENERATE an OUTLINE FROM an ORGANIZATION:

```
SASE: GEN OUT FRO ORG 1 ALL USE ALL ENT LOG
```

2. To GENERATE an OUTLINE STEPWISE:

```
SASE: GEN OUT STE USE ALL ENT LOG
```

```
DC: L
DC: 1101
HED 1 LEV 1 : BUILDING
POTENTIAL DATUM ALONG WITH UNMATCHED ARGUMENTS
DATUM 12 ARG LEFT: NONE
DATUM 23 ARG LEFT: 1104 1105
DATUM 24 ARG LEFT: 200 202
DD: 23, 24
DC:
:
DC: END
```

ERRORS:

1. An OUTLINE exists for the current VERSION.
2. SCOPELIST does not exist in the current VERSION.
3. HIERARCHY does not exist in the current VERSION.
4. Specified CLASSIFIERS are not defined in the current VERSION.

5. Specified CLASSIFIERS are not root nodes of FIELD(s) or FACET(s).
6. Attempt to GENERATE an OUTLINE FROM an ORGANIZATION by specifying an ENTRY value of RELEVANT.

CONTINUE OUTLINE

FORMAT:

CONTINUE OUTLINE

Syntax of Dialogue:

```
User entry 1 : || number1  [, number2] ||  
                || PARENT                ||  
                || CLASSIFIER           ||  
                || END                   ||
```

```
User entry 2 : || number3  (, number3) ||  
                || (Null)                ||
```

Explanation of Terms:

number1 - classifier reference number
number2 - level number
number3 - provision (i.e. datum) reference number
(Null) - blank line or carriage return

FUNCTION:

CONTINUE OUTLINE is applicable only if the STEPWISE option was initially specified in GENERATE OUTLINE. This command CONTINUES the previous OUTLINE dialogue session (See GENERATE OUTLINE). To assist the user, SASE first prints out the last input entry of the current OUTLINE. Following this, a system prompt of "DC:" appears signaling the user to enter any additional input.

RESTRICTIONS:

1. The OUTLINE must exist in the current VERSION.
2. See the RESTRICTIONS of GENERATE OUTLINE.

EXAMPLES:

1. To CONTINUE an existing OUTLINE:

```
SASE: CON OUT
LAST ENTRY WAS
HED 9 LEV 10 : MATERIAL
DC: 1101, 11
HED 10 LEV 11 : BUILDING
DD: 108
DC:
:
:
:
DC: END
SASE:
```

ERRORS:

1. Attempt to access an undefined OUTLINE.
2. Specified CLASSIFIERS are not defined.
3. Specified CLASSIFIERS are not root nodes of FIELD(s) or FACET(s).

DELETE OUTLINE

FORMAT:

DELETE OUTLINE

FUNCTION:

The existing OUTLINE is DELETED. Prior to DELETion, the system will prompt the user for confirmation.

RESTRICTIONS:

1. The OUTLINE must exist in the current VERSION.

EXAMPLES:

1. To DELETE an OUTLINE

```
SASE:      DEL OUT
** CONFIRM TO DELETE THE ENTIRE [OUT] OF THE CURRENT
   VERSION. [Y/N] ? Y
** DELETION COMPLETED **
```

ERRORS:

1. Attempt to access an undefined OUTLINE.

DISPLAY OUTLINE

FORMAT:

```

DISPLAY OUTLINE || STRUCTURE || || || ALL ||
                 || LIST || ALL || || || REQUIREMENT ||
                 || (classifier || || || DETERMINATION ||
                 || labels) || || ||
                 || ALL* || || [PRINTER] ||
                 || (datumlabels) ||
    
```

Explanation of Terms:

```

classifierlabels ::= { REFERENCE
                     NAME
                     TITLE
                     TYPE
                     PARENT
                     FOSTER }
    
```

```

datumlabels ::= { REFERENCE
                  NAME
                  TITLE
                  SECTION
                  PAGE
                  VALUE
                  SOURCE
                  TYPE
                  STATUS
                  UTILIZATION
                  INGREDIENTS
                  ARGUMENTS
                  EQUIVALENTS }
    
```

*Note: If STRUCTURE is used, then REFERENCE and TITLE are the only recognizable datum labels that will be displayed.

FUNCTION:

The current OUTLINE is DISPLAYed.

DISPLAY OUTLINE STRUCTURE is used to DISPLAY CLASSIFIERS and their corresponding DATUMs in a STRUCTURED form (indentation is used). DISPLAY OUTLINE LIST is used to DISPLAY a tabular form of the output. If DISPLAY OUTLINE LIST ALL is used, ALL existing attributes of each heading CLASSIFIER are DISPLAYed. Otherwise, only the attributes corresponding to the indicated CLASSIFIER labels are DISPLAYed.

If the second ALL is used, ALL provisions (data items) under the heading are DISPLAYed. Otherwise, only those provisions labeled as REQUIREMENT or DETERMINATION are DISPLAYed.

If the third ALL is used, ALL existing attributes of the data items shown as provisions are DISPLAYed. Otherwise, only the attributes corresponding to the indicated DATUM labels are DISPLAYed.

RESTRICTIONS:

1. OUTLINE must exist in the current VERSION .
2. In order to DISPLAY the COMMENT of the OUTLINE whenever DATUM labels are also requested, the user must type "COMMENT" twice. Otherwise, the system only DISPLAYs the COMMENT for DATUM.

EXAMPLES:

1. To DISPLAY the OUTLINE in a STRUCTURED form including ALL provisions with TITLES:

```
SASE:  DIS  OUT  STR  ALL T
Classifier          Provision
-----
BUILDING
PART OF BUILDING   TITLE
SPECIFIC BUILDING  TITLE: SECTION 1.1
SEISMIC PERFORMANCE TITLE: TABLE 1.1
CATEGORY A         TITLE: FUNCTION 1.1
CATEGORY B         TITLE: TABLE 1.2
MATERIAL NATURE    TITLE:
                   TITLE: TABLE 2.1
```

ERRORS:

1. OUTLINE does not exist in the current VERSION.
2. The DISPLAY requested contains no OUTLINE.

GENERATE PAGE

FORMAT:

GENERATE PAGE number

FUNCTION:

A PAGE number is GENERATED for use with the ENTER DATUM command. A new PAGE number must be GENERATED when a change occurs. The GENERATED PAGE number is automatically ENTERED in the DATUM description as the PAGE attribute.

RESTRICTIONS:

1. This command is applicable only when a CHAPTER is active.

EXAMPLES:

1. To GENERATE a new PAGE number 292 ENTER DATUM 1000 and referenced by PAGE 292:

```
SASE: USE CHA CHI
** (CHI) IS PUT INTO CURRENT CHAPTER.
```

```
SASE: GEN PAG 292
SASE: ENT DAT 1000 DAT1
SASE: DIS DAT 1000 PAGE
```

```
REFER PAGE
1000 292
```

ERRORS:

1. CHAPTER is not defined.

GENERATE REFERENCE

FORMAT:

GENERATE REFERENCE BASE number INCREMENT number.

FUNCTION:

REFERENCE numbers are GENERATED for use in the ENTER DATUM or ENTER CLASSIFIER commands. Beginning with the BASE number, these REFERENCE numbers are successively incremented by the amount specified in the second number.

RESTRICTIONS:

1. This command is applicable for DATUMs only when a CHAPTER is active.
2. This command is applicable for CLASSIFIERS only when a VERSION is active.

EXAMPLES:

1. To GENERATE REFERENCE numbers beginning at 1200 with increments of 5:

```
SASE: USE CHA CHI
** <CHI > IS PUT INTO CURRENT CHAPTER.
```

```
SASE: GEN R B 1200 I 5
SASE: ENT DAT NA BLDG
SASE: ENT DAT NA FLOOR
SASE: DIS DAT ALL R NA
```

```
REFER NAME
```

```
1200 BLDG
-----
```

```
REFER NAME
```

```
1205 FLOOR
-----
```

ERRORS:

1. The GENERATED REFERENCE number corresponds to a previously defined DATUM or CLASSIFIER REFERENCE number.
2. VERSION or CHAPTER is not defined.

GENERATE SECTION

FORMAT:

GENERATE SECTION name.

FUNCTION:

SECTION references are GENERATED for use in the ENTER DATUM command. GENERATE SECTION name is used to:

1. ENTER an initial SECTION name; or
2. ENTER new names when a change occurs.

The current section name is automatically ENTERed in the DATUM description as the SECTION attribute.

RESTRICTIONS:

1. This command is applicable only when a current CHAPTER is active.

EXAMPLES:

1. To ENTER a new SECTION name A3.2 referencing DATUM 1000:

```
SASE: USE CHA CHI
** <CHI > IS PUT INTO CURRENT CHAPTER.
```

```
SASE: GEN SEC A3.2
SASE: ENT DAT 1000 DAT1
SASE: DIS DAT 1000 SEC
```

```
REFER SECTI
```

```
1010 A3.2
```

ERRORS:

1. CHAPTER is not defined.

APPENDIX SASE ERROR MESSAGES

SASE generates one of the following error messages whenever a user attempts to perform an illegal operation or data entry.

- 1 STANDARD AND/OR VERSION ARE NOT DEFINED
- 2 STANDARD IS NOT DEFINED
- 3 VERSION IS NOT DEFINED
- 4 CHAPTER IS NOT DEFINED
- 5 ATTEMPT TO CREATE INFORMATION ALREADY DEFINED
- 6 ATTEMPT TO USE INFORMATION THAT IS NOT DEFINED
- 7 LIST IS EMPTY
- 8 ATTEMPT TO ADD INFORMATION ALREADY DEFINED
- 9 ATTEMPT TO DELETE INFORMATION THAT IS NOT DEFINED
- 10 ATTEMPT TO MODIFY INFORMATION THAT IS NOT DEFINED
- 11 REFERENCE NUMBER IS OUT OF RANGE
- 12 ATTEMPT TO ENTER A DATUM ALREADY DEFINED
- 13 ATTEMPT TO WORK WITH A DATUM THAT IS NOT DEFINED
- 14 THE SAME NUMBER IS ENTERED
- 15 NO TEXTUAL EXPLANATION FOR THIS ERROR
- 16 REFERENCE MADE TO A DATUM/CLASSIFIER THAT IS NOT DEFINED
- 17 OUTPUT TO BE DISPLAYED IS EMPTY
- 18 ATTEMPT TO DISPLAY INFORMATION THAT IS NOT DEFINED
- 19 ATTEMPT TO ENTER A FUNCTION ALREADY DEFINED
- 20 ATTEMPT TO WORK WITH A FUNCTION THAT IS NOT DEFINED
- 21 DATUM IS NOT DEFINED
- 22 DATUM'S TYPE IS NOT FUNCTION
- 23 ATTEMPT TO ENTER A TABLE ALREADY DEFINED
- 24 A TREE IS ALREADY GENERATED OR SAVED
- 25 ATTEMPT TO WORK WITH A TABLE THAT IS NOT DEFINED
- 26 DATUM'S TYPE IS NOT TABLE
- 27 ATTEMPT TO ENTER DATUM ALREADY SAVED IN BACKUP
- 28 ATTEMPT TO CREATE INFORMATION ALREADY SAVED IN BACKUP
- 29 NULL ACTION DETECTED IN TABLE
- 30 NO TEXTUAL EXPLANATION FOR THIS ERROR
- 31 NO TEXTUAL EXPLANATION FOR THIS ERROR
- 32 NO TEXTUAL EXPLANATION FOR THIS ERROR
- 33 ENTRIES ARE NOT ENTERED BY EITHER CONDITION OR RULE
- 34 ATTEMPT TO WORK WITH AN ORGANIZATION THAT IS NOT DEFINED
- 35 CONDITION NUMBERS ARE NOT SEQUENTIAL
- 36 SIZE OF COND ENTRIES IS NOT THE SAME AS PREVIOUS SIZE
- 37 ACTION NUMBER IS NOT SEQUENTIAL
- 38 RULE NUMBER IS NOT DEFINED
- 39 RULE'S ACTION IS ALREADY DEFINED
- 40 SIZE OF RULE'S ENTRIES IS NOT THE SAME AS PREVIOUS ONE
- 41 ACTION NUMBER IS NOT DEFINED
- 42 RULE NUMBER IS NOT SEQUENTIAL
- 43 CONDITION NUMBER IS NOT DEFINED
- 44 ATTEMPT TO MODIFY CONDITION'S STUB THAT IS NOT DEFINED
- 45 ATTEMPT TO ADD CONDITION'S STUB ALREADY DEFINED

APPENDIX SASE ERROR MESSAGES

46 ATTEMPT TO MODIFY CONDITION ENTRY THAT IS NOT DEFINED
47 ATTEMPT TO ADD CONDITION ENTRY NOT FULLY DEFINED
48 ATTEMPT TO MODIFY AN ACTION STUB THAT IS UNDEFINED
49 ATTEMPT TO ADD AN ACTION STUB THAT IS ALREADY DEFINED
50 NO TEXTUAL EXPLANATION FOR THIS ERROR
51 ATTEMPT TO ADD A RULE ACTION THAT IS ALREADY DEFINED
52 NO TEXTUAL EXPLANATION FOR THIS ERROR
53 ATTEMPT TO ADD RULE ENTRIES THAT ARE FULLY/PARTIALLY DEFINED
54 ATTEMPT TO DELETE CONDITION STUB THAT IS NOT DEFINED
55 ATTEMPT TO DELETE CONDITION ENTRY THAT IS NOT DEFINED
56 DATUM'S INGRE/EQUIV CAN'T BE THE DATUM ITSELF
57 ATTEMPT TO DELETE ACTION STUB THAT IS NOT DEFINED
58 ATTEMPT TO DELETE ACTION WHICH IS NOT TAKEN BY ANY RULE
59 ATTEMPT TO DELETE THE LAST RULE
60 ATTEMPT TO DELETE RULE ENTRY THAT IS NOT DEFINED
61 AN INDEX HAS BEEN SAVED PREVIOUSLY
62 ATTEMPT TO DELETE RULE ACTION THAT IS NOT DEFINED
63 ATTEMPT TO DELETE THE LAST CONDITION
64 ATTEMPT TO DELETE THE LAST ACTION
65 ATTEMPT TO RESEQUENCE SINGLE CONDITION, ACTION OR RULE
66 NUMBER MISSING IN RESEQUENCING TABLES CON, ACT OR RULE
67 TABLE'S ENTRIES ARE NOT FULLY DEFINED
68 NO TEXTUAL EXPLANATION FOR THIS ERROR
69 REDUNDANT OR CONTRADICTORY RULES AS SHOWN BELOW
70 THERE ARE REDUNDANT CONDITIONS
71 THERE ARE SOME GIVEN RULES NOT REACHED IN THE TREE
72 CAN'T FIND A PROPER NODE TO START TRACING A LOOP IN THE NETWORK
73 CLASSIFIER IS NOT DEFINED
74 CLASSIFIER IS ALREADY DEFINED
75 CONFLICT BETWEEN CLASSIFIER'S TYPE AND PARENT/FOSTER
76 TYPE IS ACT (TRA), BUT ATTEMPT TO ADD FOS (PAR)
77 CLASSIFIER, PARENT, FOSTER AND CONTEXT ARE NOT DISTINCT
78 ATTEMPT TO ADD CONTEXT ALREADY DEFINED
79 EVERY DATUM OF THE NETWORK TO BE GENERATED HAS INGREDIENT(S)
80 ATTEMPT TO DELETE/MODIFY FOSTER THAT IS NOT DEFINED
81 ATTEMPT TO MODIFY FOSTER'S CONTEXT THAT IS NOT DEFINED
82 A NETWORK HAS BEEN PREVIOUSLY SAVED
83 ATTEMPT TO ADD THE COMMENT WHICH IS ALREADY ENTERED
84 ATTEMPT TO MODIFY A COMMENT THAT IS UNSPECIFIED
85 A NETWORK HAS NOT BEEN SAVED PREVIOUSLY
86 NO DATUM DEFINED FOR THE CURRENT CHAPTER/VERSION
87 NO INGREDIENTS IN THE CURRENT CHAPTER/VERSION
88 A NETWORK'S DATUM WAS IN THE DATABASE, BUT HAS BEEN DELETED NOW
89 ROOT NODE IS NOT IN THE NETWORK
90 A HIERARCHY HAS BEEN PREVIOUSLY SAVED
91 THE OBJECT HAS NOT BEEN SAVED OR ENTERED PREVIOUSLY
92 NO CLASSIFIERS DEFINED FOR THE CURRENT VERSION
93 CLASSIFIERS ABOVE HAVE BOTH PARENT AND FOSTER
94 A HIERARCHY LOOP ABOVE IS DETECTED

APPENDIX SASE ERROR MESSAGES

95 CLASSIFIERS/FACETS SHOWN ABOVE ARE ISOLATED/DETACHED
96 FIELDS ARE NOT GENERATED
97 NO SINGLE FIELD EXISTS, ALL FACETS
98 DATUM/CLASSIFIER WAS STORED IN DATA BASE, BUT HAS BEEN DELETED NOW
99 FIELD/FACET NUMBER GIVEN IS NOT A ROOT NODE
100 A SCOPELIST HAS BEEN SAVED PREVIOUSLY
101 HIE, SCO, IND, AND OUTL TO BE GENERATED FROM THE SAME VERSION
102 A HIERARCHY HAS NOT BEEN SAVED PREVIOUSLY
103 CHAPTER(S) SPECIFIED DO NOT CONTAIN ANY DATUM
104 DATUM OF CHAPTER(S) SPECIFIED DO NOT HAVE ARGUMENTS
105 ATTEMPT TO DEL/MOD DATUM'S TYPE WHILE ITS FUN/TAB EXISTS
106 THE TREE HAS NOT BEEN SAVED PREVIOUSLY
107 TOO MANY NODES (120) IN THE TREE TO BE PROCESSED
108 TOO MANY LEVELS (16) OF TREE'S TOPOLOGY TO BE PROCESSED
109 TOO MANY DATUM INVOLVED (1250) TO BE PROCESSED
110 DATUM (CLAS) ARE INCONSISTENT (DELETE AND REENTER)
111 RECORDS WITH SAME REF NUMBER EXIST IN DATA BASE (DELETE AND REENTER)
112 TOO MANY ARGUMENTS INVOLVED (2500) TO BE PROCESSED
113 TOO MANY CLASSIFIERS INVOLVED (300) TO BE PROCESSED
114 TOO MANY FOSTERS INVOLVED (100) TO BE PROCESSED
115 TOO MANY CONTEXTS INVOLVED (100) TO BE PROCESSED
116 TOO MANY DATUMS IN THE NETWORK (1250) TO BE PROCESSED
117 TOO MANY BRANCHES (2200) TO BE PROCESSED
118 THERE IS NO CLASSIFIER LEFT IN INDEX/OUTLINE TO BE GEN/DIS
119 AN ORGANIZATION HAS BEEN ENTERED PREVIOUSLY
120 THE HEADING SPECIFIED DOES NOT EXIST
121 ORGANIZATION INFO IN DATA BASE IS INCONSISTENT(DELETE AND REENTER)
122 THE CLA OF LAST ENTRY WAS IN DATA BASE, BUT HAS BEEN DELETED NOW
123 THE MOVE SPECIFIED IS AMBIGUOUS OR UNNECESSARY
124 AN OUTLINE HAS BEEN SAVED PREVIOUSLY
125 GENERATING OUTLINE IMPLICITY HAS NOT BEEN IMPLEMENTED
126 THE ORGANIZATION IS EMPTY
127 ENTRY RELEVANT IS NOT ALLOWED IN GEN OUT FROM ORG
128 STEPWISE GENERATED OUTLINE HAS NOT BEEN SAVED PREVIOUSLY
129 THE CLASSIFIER ENTERED WAS DROPPED
130 THE CLASSIFIER ENTERED HAS NO SON
131 THE NUMBER ENTERED DOES NOT BELONG TO ANY OF THE POTENTIAL DATUMS
132 THE CLASSIFIER ENTERED IS NOT INCLUDED IN THE OPERATION AS YET