ERS for Source Code Utility (NOS 170 Version)

03/24/80 REV: J

EXTERNAL REFERENCE SPECIFICATION

FOR

C180 SOURCE CODE UTILITY

NOS 170 Prototype Version

Submited: _____

Disclaimer:

This document is an internal working paper only. It is subject to change, and does not necessarily represent any official intent on the part of CDC.

03/24/80 REV: J

ERS for Source Code Utility (NOS 170 Version)

REVISION DEFINITION SHEET

REVDATEDESCRIPTIONA08/10/76Version 2.0 _ original release.B11/15/76Version 2.1 _ NOS conversion.C08/15/77Update for release under 3.3 of SES Subsystem.D06/30/78Version 3.0 - Re-direction.E12/04/78Version 3.1 - Utility re-written in CYBL, dependence on SES Subsystem removed and capabilities enhanced. ERS rewritten.F02/06/79ERS revised in response to DCS comments.G05/11/79ERS revised in response to DCS comments.H11/06/79Temporary revision of ERS in response to DCS comments etc.J03/24/80This revision describes the NOS 170 prototype implementation of the utility in particular. Inputs from DCS comments and design team meetings have been included.			1	
 B 11/15/76 Version 2.1 NOS conversion. C 108/15/77 Update for release under 3.3 of SES Subsystem. D 106/30/78 Version 3.0 - Re-direction. E 12/04/78 Version 3.1 - Utility re-written in CYBIL, dependence on SES Subsystem removed and capabilities enhanced. E E 12/06/79 ERS revised in response to DCS comments. F 102/06/79 ERS revised in response to DCS comments. H 11/06/79 Temporary revision of ERS in response to DCS comments etc. J 03/24/80 This revision describes the NOS 170 prototype implementation of the utility in particular. Inputs from DCS comments and design feam 	r 1 : 1 • .	DESCRIPTION	DATE	REV
 C 108/15/77 Update for release under 3.3 of SES Subsystem. D 106/30/78 Version 3.0 - Re-direction. E 12/04/78 Version 3.1 - Utility re-written in CYBIL, dependence on SES Subsystem removed and capabilities enhanced. ERS rewritten. F 102/06/79 ERS revised in response to DCS comments. G 105/11/79 ERS revised in response to DCS comments. H 11/06/79 Temporary revision of ERS in response to DCS comments etc. J 03/24/80 This revision describes the NOS 170 prototype implementation of the utility in particular. Inputs from DCS comments and design team 	, 	Version 2.0 _ original release.	08/10/76	А
 Subsystem. D 06/30/78 Version 3.0 - Re-direction. E 12/04/78 Version 3.1 - Utility re-written in CYBIL, dependence on SES Subsystem removed and capabilities enhanced. ERS rewritten. F 02/06/79 ERS revised in response to DCS comments. G 05/11/79 ERS revised in response to DCS comments. H 11/06/79 Temporary revision of ERS in response to DCS comments etc. J 03/24/80 This revision describes the NOS 170 prototype implementation of the utility in particular. Inputs from DCS comments and design feam) 	Version 2.1 _ NOS conversion.	11/15/76	B
 E 12/04/78 Version 3.1 - Utility re-written in CYBIL, dependence on SES Subsystem removed and capabilities enhanced. ERS rewritten. F 02/06/79 ERS revised in response to DCS comments. G 05/11/79 ERS revised in response to DCS comments. H 11/06/79 Temporary revision of ERS in response to DCS comments etc. J 03/24/80 This revision describes the NOS 170 prototype implementation of the utility in particular. Inputs from DCS comments and design team 	 		08/15/77	С
 CYBIL, dependence on SES Subsystem removed and capabilities enhanced. ERS rewritten. F 02/06/79 ERS revised in response to DCS comments. G 05/11/79 ERS revised in response to DCS comments. H 11/06/79 Temporary revision of ERS in response to DCS comments etc. J 03/24/80 This revision describes the NOS 170 prototype implementation of the utility in particular. Inputs from DCS comments and design team) 	Version 3.0 - Re-direction.	06/30/78	D
G 05/11/79 ERS revised in response to DCS comments. H 11/06/79 Temporary revision of ERS in response to DCS comments etc. J 03/24/80 This revision describes the NOS 170 prototype implementation of the utility in particular. Inputs from DCS comments and design team		I CYBIL, dependence on SES Subsystem I removed and capabilities enhanced.	12/04/78	E
H 11/06/79 Temporary revision of ERS in I response to DCS comments etc. J 03/24/80 This revision describes the NOS 170 I prototype implementation of the Utility in particular. Inputs from DCS comments and design team			02/06/79	F
I 1 response to DCS comments etc. 1 I 1 1 1 I 1 1 1 I 1 1 1 I 03/24/80 This revision describes the NOS 170 1 I 1 prototype implementation of the I 1 I 1 utility in particular. Inputs from I 1 I DCS comments and design team 1			05/11/79	G
II prototype implementationoftheIII utilityIn particular.Inputs fromIII DCScommentsanddesignteam		•	11/06/79	н
		l prototype implementation of the l utility in particular. Inputs from l DCS comments and design team	03/24/80 1 1	J
1 1 1 1]]] [8 3 1 1	1 1 1 1 1 1	
· · · · · · · · · · · · · · · · · · ·)] •	, +	1] ********	

c COPYRIGHT CONTROL DATA CORPORATION 1980 All Rights Reserved

COMPANY PRIVATE

ł

ŧ

1-1

	•	170 Version)	REV: J
PREFACE			

1.0 PREFACE

1.1 SCOPE OF DOCUMENT

This document is intended to contain the information necessary for use of the NOS 170 prototype version of SCU by a terminal or batch user. Since there are some features which are of necessity operating system or machine dependent and because not all features which will appear in the NOS/VE implementation will be present in the prototype these two implementations are being described in separate documents. The revision bars shown in this document have been generated relative to the ERS For Source Code Utility DCS number ARH1766 revision G. That was the last ERS submitted to DCS which attempted to describe both implementations. An ERS describing the NOS/VE version of the utility will be submitted to DCS under another number.

1.2 APPLICABLE DOCUMENTS

Document_	DCS Number
SES User's Handbook	ARH1833
NOS Reference Manual	
NOS/VE ERS - Command Interface	ARH3609
CYBER 180 System Interface Standard	S2196

33333444444

7 8

9 10

111

112

118

114

115

116

117

118

119

120

121

125

- 45 46 47
- 48

COMPANY PRIVATE

							00764700
ERS	for	Source	Code	Utility	(NOS 170	Version)	REV#J
~~~~		***				ه بد به	a day any any any any any any any any any a

### 2.0 INTRODUCTION

2.0 INTRODUCTION

The purpose of this document is to describe the NOS 170 prototype version of a utility which will provide for introduction and maintenance of ASCII source data on the Cyber 180. While emphasis is placed on the control and documentation necessary in large development groups, it can be used by individuals maintaining private source libraries.

The utility described below includes an interactive editor allowing for interactive development of new source units or corrections to existing source units. It provides multi_deck and build capabilities, allows for grouping of source decks and correction sets into larger identifiable units and associates descriptive information with source decks and correction sets.

The commands described in this document make use of the System Command Language (SCL) syntax which has been defined for NOS/VE, the operating system that will execute on Cyber 180.

This document describes the implementation of this utility on NOS 170. The reader should be aware that some of the parameters described below are machine dependent and will require adjustment when the final version of this utility is implemented to execute under NOS/VE. The most common examples of parameter information that will change is the length of user names and file names. Where possible the NOS 170 version will be made to emulate the final environment.

### 2.1 CONCEPTS

This section introduces terms used throughout laten sections of this document.

### 2.1.1 SOURCE LIBRARY

A source library is a file containing a collection of 45 source decks. A source library will usually represent all or 46 part of the source code for a product at a certain stage of 47 development. 48

456

1 7

8

9

10

11

12 13

114

115

116

117

118

119

21

122

128

25 26

127

128

29

38

31

32

33

38

39

48 41 42

43

COMPANY PRIVATE

### 03/24/80

2-2

12

3

6

5

6

7

8

9

10

11 12 13

14 15

16

17

:18

119

120 21 22

> 23 24

25

26

27

28

29

30

31

32

33

34

35 36 37

38 39

48

4 T

42

43

44

45

46

ERS for Source Code Utility (NOS 170 Version) REV:J

### 2.0 INTRODUCTION 2.1.2 SOURCE DECK

### 2.1.2 SOURCE DECK

A source deck is a physical collection of lines of source text. The text may consist of source code, documentation on any other type of source data. SCU text embedded directives may be intermixed with the source data. Each line of text will have associated with it a unique (with respect to the deck) line identifier which can be used to identify the line. A line identifier consists of an alphanumeric modification name and a sequence number relative to that modification as it applies to a given deck.

### 2.1.3 GROUP

A group is a name associated with a collection of decks to allow for the logically related decks to be extracted from a source library together or merely to document the fact that they are logically related. A deck may have up to 255 group names associated with it.

### 2.1.4 MODIFICATION

In this document the term modification will be used to mean the changes introduced to a deck or series of decks by an edit session. It is possible to have an edit session be considered a continuation of a previous one. Each modification has associated with it a unique user assigned name. This name will be used to form the line identifier of each source line introduced by the modification.

Modifications are stored on the source library by SCU in terms of their effect on the source decks to which they apply, rather than in the form of the original commands that introduced the change.

### 2.1.5 FEATURE

A feature is a name associated with a modification. Associating the same feature name with several modifications to the same or different decks allows one the ability to group modifications together to be applied or ignored as a group. For example all the modifications necessary to make a major change in an interactive interface to the system could be associated with a common feature name. A modification may have only one feature name associated with it.

.

2-3

		03/24/80
RS for Source Code Util	ITY (NUS 1/U Version)	REV: J
•0 INTRODUCTION •1.6 STATES		
2.1.6 STATES		
The state reflects th	has a specific state as e degree of maturity the r ing states are defined:	
0 – Experimental	should probably not be build by anyone except	e included in a t the person it or someone
1 - Developmental	Code which has been unit	tested.
2 - Stable	Code which has passed re	egression tests.
3 - Verified	Code which has been found advertised.	d to perform as
4 - Released	Code which has been co and approved.	ompletely tested
information altered.	state 0 may have its text Also if the user has p e of a state 0 modification	roper authority;
state or lowered t	tates 1, 2 or 3 may only o state 0. No altera on is allowed for modifica	tion of text or
description or state.	n state 4 cannot be a In other words, errors te 4 modification must be name.	in text lines
2.1.7 AUTHORITY		
user name that is manipulations on a so associated with it a necessary to change authorities specified - Verified and 4	he term authority will be validated by SCU to a urce librbary. A source specification of each of the state of a modi- are for 1 - Developmental - Released. Each modi- ch may include any of	perform certain ce library has four authorities fication. The l, 2 - Stable, 3 fication has a

COMPANY PRIVATE

03/24/80

主

2

6 7

8

9

10

11

12

13

14

15

16 17 18

119

21

22

23

24

25

26 127

128

129

130 31 32

> 33 34

> 35

36

37

38 39 48

41 42

1345

ERS	for	Source	Code	Utility	(NOS	170	Version)	REV#J
	* ** ** ** *		~~~~~				جاد جاد جاد ماه جد جاد جاد خاد جاد خاد ک	

### 2.0 INTRODUCTION 2.1.7 AUTHORITY

Experimental. A user must have the specified authority for a library (or a higher one) to raise the state of a modification to a given level or to lower it from that state to state 0.

### 2.1.8 CRITERIA FILE

A criteria file contains directives used to limit which features and modifications at which states are to be included when blocks of source data are expanded from a source library. Similarly decks or groups may be included on excluded through directives on a criteria file. In addition values may be declared in a criteria file which can be used to trigger conditional inclusion of blocks of source data when decks are extracted from a source library and expanded for a processor.

### 2.1.9 TEXT EMBEDDED DIRECTIVES

When a user wishes to prepare source data for a processor, SCU will expand the source under the control of text embedded directives some of which interact with the contents of a criteria file. Text embedded directives are intermixed with the source data. Text embedded directives are recognized as distinct from source data by the occurrence of a specific character in the first postion on the line followed immediately by a keyword. In this document asterisk is shown as the key character in all examples. Some examples of text embedded directives are *COPY and *ELSE.

### 2.1.10 NORMAL SEARCH ORDER

Some SCU commands working with decks allow the use of multiple source libraries. In locating a deck, the source libraries are searched in the order in which they are given in the list.

### 2.1.11 BASIC STATUS CONCEPTS

System Command Language (SCL) under NOS/VE will require all43commands to have a status parameter. This parameter will be44used to try to standardize the processing of error messages45and abnormal responses from command processors. In the146version of this utility which is implemented to execute under147NOS on CYBER 170 systems the value of this parameter must be148

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 2.0 INTRODUCTION 2.1.11 BASIC STATUS CONCEPTS given as the name of one of the job control regelsters, R1, 1 1 R2, R3 or EF. The named register will be set non-zero if an 1 2 error is encountered while processing a command. If R1, R2 or 1 8 R3 is given, the value entered into the register will 1 6 correspond to the condition codes shown in the section on 1 5 1 6 messages. 7 8 9 2.1.12 NAMES 10 1**Ť** Names used in SCU will follow the NOS/VE patterns. 12 Valid characters to be included in a name are upper and lower case 13 letters, the characters "#", "\$", "@" and "_" (underline) and 14 decimal digits. Names must not begin with a digit. Library 15 names, deck names, feature names and group names will contain 16 1 to 31 characters. The only exceptions to the pattern will 17 be modification names which will be restricted to a maximum of 18 19 9 characters in length. Modification sequence numbers will be 20 6 digits long and range from 1 to 262143. 21 22 2.1.13 STRINGS 123 24 Some parameters for the SCU commands are described as being 125 strings. System Command Language syntax requires strings to 126 127 be enclosed in single quotes. 28 29 2.1.14 LOCAL FILES 30 31 SCU assumes all files it is to deal with are local with the 132 exception of those named on the BASE and RESULT parameters 133 below. The user may specify the user name for permanent 134 files, if different than his own, by including it as the 135 second value in a value set. It is the user's responsibility 136 to acquire access to other files prior to initiating SCU 137 commands to use them. 138 The NOS 170 version of SCU uses a series of scratch files 139 internally whose names begin with the string "SESSC". The 140

user of the NOS 170 version should avoid making explicit use

of files with names fitting this pattern.

- 46 47
- 48

1234567

8

110

111

112

113

114

115

16

18

19

20

21

22

23

24

25

25

27

28

29

139

131

132

133

134

135

37

38

39 40 41

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J

### 3.0 COMMANDS

*****

3.0 COMMANDS

The commands listed below are available in the SCU environment. Commands for the editor, text embedded directives and criteria file directives are described later. The commands listed here function as separate job control statements and are intended to use a syntax as similar as possible to that of the System Command Language (SCL) for NOS/VE. When accessed on NOS 170. these commands must be prefixed by the characters "scu.". The commands can also be bound together in a more intimate fashion as described below under command grouping.

Parameter defaults are assigned as follows. Required parameters where possible will have static defaults assigned initially such as BISE for the name of the base source library. The commands listed here are presented in a manner so as to emphasize the parameters of the commands. Optional parameters are enclosed with square brackets. As with NOS/VE SCL commands, commas and/or spaces can be used to separate parameters. Parameters can be given positionally, by keyword=value or a mixture. In the latter case specification of a parameter by keyword "resets" the positional pointer to the following parameter field. Where available, shortened forms for the keywords are shown. An ellipsis (two or more periods) at the end of a line indicate that the command is being continued on the next line. On NOS 170 lower case letters appearing in strings on the first physical line of an SCU command will be capitalized. To avoid this unpleasant side affect of executing on NOS 170 a user should include strings which are parameters to SCU on continuation lines.

As on NOS/VE all lower case letters appearing inside names (and references to names) are translated to their upper case counterparts by the SCL interpreter.

The short form of each command is listed with the command.

43 44

- 45
- 46
- 47

DC SOFTWARE ENGINEERING SYSTEM	
RS for Source Code Utility (NOS 170 Version)	03/24/80 REV: J
na n	-
• 1 SOURCE LIBRARIES	
	یون میک بیش بین مید میک بین بین بین بین میک میک بین میک بین ایک بین ایک بین ایک ایک ایک ایک ایک ایک ایک ایک ای
3.1 SOURCE_LIBRARIES	
The following commands are available for manipulating source litraries.	creating and
3.1.1 CREATE_LIBRARY 1 CRL	
This command creates the header of a source li	brary on the
specified file and includes in this header th	
information about the library including the libra character for text embedded directives and use	
assigned authorities.	TIGHICS IV DE
create_library [name= <library_name>]</library_name>	
[description= <descriptive_string>] [authority_1=<user_name>]</user_name></descriptive_string>	1
[author Ity_2= <user_name>]</user_name>	
[authorIty_3= <user_name>]</user_name>	
[authority_4= <user_name>]</user_name>	
[key= <char>] [version=<version_string>]</version_string></char>	
[result= <file_name>]</file_name>	
[status=r11r21r31ef]	
name 1 na : This specifies the name of	f the source
library. If this parameter is omitted th	
result file will be used.	
description 1 de : This 1 to 256 character s	tring is used
to describe the source code maintained on the	_
authority_1 : a1 : This is the 7 character have authority 1 (developmental authorit	
library. If parameters authority_1 through	
are not given, the user name that the	
executing under will be used.	
authority_2   a2   This is the 7 character	user name to
have authority 2 (stable authority) for this	
authority_3 : a3 : This is the 7 character (	
have authority 3 (verified authority) over	THIS LIDEARY.
authority_4 : a4 : This is the 7 character u	iser name to
have authority 4 (released authority) for th	nis library.
	COMPANY PRIVATE
le la	VUICHINE CRIVER

#### CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 3.0 COMMANDS 3.1.1 CREATE_LIBRARY I CRL 1 key : k : This one character string specifies the key 2 character used to flag text embedded directives on this 3 source library. This defaults to *. 6 5 : 6 version 1 ve : This 1 to 31 character string is used to give a version number for the library. 1 7 8 result 1 r : This is the name of the file on which the : 9 110 source library is to be created. RESULT is used as a default value. 11 12 status I st : See basic status concept. 13 14 Examples: 115 scu.create_library scu_source_code "Neat stuff." a4=m1p2523 .. 116 r=scupl 17 118 SCU.CRL Table_Tennis_League_Records 119 20 In the first example the library name and description 121 parameters are given positionally and the authority_4 and 122 result file names are given with the keyword = value syntax. 123 25 The remaining authorities would default to the user name the command was executed under and * would be used to flag text 25 embedded directives. 26 127 In the second example only the library name is given 128 explicitly and a source library will be created on the file RESULT. :29 30 31 32 33 34 35 36 38 38 39 40 41 42 43 45 45 46 47

ŝ.

COMPANY PRIVATE

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 3.0 COMMANDS 3.1.2 DISPLAY_LIBRARY : DIL 3.1.2 DISPLAY_LIBRARY 1 DIL 1 2 This command lists the descriptive information stored in 3 the source library header and optionally the contents of deck, 4 group, modification and feature lists for the library. 5 6 display_library [base=<file_name>] : 7 [list=<local_file_name>] 1 8 [brief | full] 1 9 [status=r1!r2!r3!ef] 111 11 base 1 ba : This is the name of the file containing the 112 base source library. If this parameter is not given an 113 attempt is made to access a file with the name BASE. 114 15 list ! I : This is the name of the file which receives 16 the listing. If this parameter is not given, the file 17 name OUTPUT will be used. 18 19 brief 1 br 1 full 1 fu : If the command contains the 28 keyword BRIEF, only the information from the library 21 header will be displayed. If the keyword FULL is 22 present, the deck, group, modification and feature lists 23 for the library will be displayed as well. If neither is 24 present BRIEF is assumed. 25 26 status 1 st 1 See basic status concept. 27 28 Example: 129 scu.display_library ba = oldpl 130 31 This command will display only the descriptive information 32 from the library header of the source library on file OLDPL. 38 The display will be written to the file OUTPUT. 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

COC SOFTHADE ENCINEEDING SYSTEM	3-3
CDC SOFTWARE ENGINEERING SYSTEM	03/24/80
ERS for Source Code Utility (NOS 170 Version)	REV:J
	الين المراجع ا
3.0 COMMANDS	,
3.1.3 CHANGE_LIBRARY 1 CHL	
3.1.3 CHANGE_LIBRARY 1 CHL	1
This command allows a user to alter fields in description.	the library 3 4
	5
change_library [name= <library_name>]</library_name>	5
[description= <descriptive_string>]</descriptive_string>	7
[author ity_1= <user_name>]</user_name>	8
[authority_2= <user_name>]</user_name>	9
[authority_3= <user_name>]</user_name>	10
[authority_4= <user_name>]</user_name>	11 12
[version= <version_string>] [base=<file_name>]</file_name></version_string>	12
[result= <file_name>]</file_name>	114
[status=r11r21r3lef]	115
1210102-1211211010101	16
name 1 na : This specifies the name ( library.	of the source 17 18
description 1 de : This 1 to 256 character s to describe the source code maintained on th	
	22
authority_1 1 a1 : This is the 7 character	
have authority 1 for this library. In ord	
fields authority_1 through authority_4, this	
be executed under the user name tha	
corresponding authority or a higher one.	27
authority_2 1 a2 # This is the 7 character	user name to 29
have authority 2 over this library.	
have during thy 2 over this tiplary.	31
authority_3 1 a3 : This is the 7 character.u	
have authority 3 over this library.	33
	34
authority_4 1 a4 # This is the 7 character	user name to 35
have authority 4 for this library.	36
	37
version 1 ve : This 1 to 31 character string	
give a version number for the source library	
hope I be I This is the same of the file of	40 Antoina tha 14
base 1 ba : This is the name of the file c base source library. If this parameter is n	
attempt is made to access a file with the na	
attempt to mane in access a trie writh the Ha	
result i r : This is the file to receive	
library. If this parameter is not given,	
written to a file named RESULT.	147
	48

### CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 3.0 COMMANDS 3.1.3 CHANGE_LIBRARY : CHL status 1 st : See basic status concept. Example: 1.3 scu.change_library version=*1.0* ba=mylib - 6

12 23

4 5

6

7

8 9

10

111

112

113

115

116

117

119

120

121

123

124

:25

126

128

130

131

132

133

134

136

137

138

140

141

:42

143 44

ERS	for	Source	Code	Utility	(NOS	170	Version)	REV:J

### 3.0 COMMANDS

### 3.1.4 COMBINE_LIBRARY & COL

3.1.4 COMBINE_LIBRARY 1 COL

This command merges the decks from a list of source libraries into a single result library. The result library will include one deck by every name encountered on any of the source libraries. If decks by the same name occur on more than one library the deck from the library listed first will be selected. Decks will have the following logical order on the result library:

- All decks with names occurring on the last named source library will occur first in their original order.
- 2. Next the decks with names occurring on the next to last named library will occur in the order they occurred on that library.
- Decks with names occurring on the remaining libraries will occur in turn in the order they occurred on those libraries.

All source libraries must have the same key character. The result source library will have the same authorities, description, etc. as the last named source library. This command will indicate an error if there are any interlock violations (see discussion of interlock under EXTRACT_LIBRARY).

combine_library base=<file_name> [result=<file_name>] [list=<local_file_name>] [brief 1 full] [status=r11r21r31ef]

base I ba : This is a list of names of files containing source libraries to be combined. The names must be enclosed in parentheses and separated by commas.

result 1 r : This is the file to receive the combined source library. If this parameter is not given, the combined source library will be written to a file named RESULT.

list11This is the name of the file which receives145the listing.The list shows from which source library146decksonthe result file were taken.If this parameter147is not given, the file name OUTPUT will be used.148

COMPANY PRIVATE

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS 3.1.4 COMBINE LIBRARY 1 COL brief 1 br 1 full 1 fu : These keywords can be used to 1 2 select a shorter or more detailed listing. 1 3 Currently both fistings are the same. 1 - 5 status 1 st : See basic status concept. 1 6 Example: 1 8 scu.combine_library (changed, new, base) newbase 1 9 10 In this example two source libraries on files named CHANGED 11 and NEW were combined with one on a file called BASE to form a 12 result on a file named NEWBASE. To illustrate how decks are 13 selected and ordered during the processing of this command 14 (and give an example of how this command is intended to be 15 used) let us assume that the three libraries being combined 16 each contained the following decks: 17 18 CHANGED A 4 8 1 4 1 4 ~ . ~ ~ 19 OM CU 28 21 JI MI 22 80 23 24

HANGED	NEW	BASE	NEWBASE	Deck Taken Fro
URT	SHARON	JIM	JIM	CHANGED
IM	ED	BCB	BOB	CHANGED
IKE		MIKE	MIKE	CHANGED
08		BILL	BILL	BASE
		SHERMAN	SHERMAN	BASE
		CURT	CURT	CHANGED
		JOHN	JOHN	BASE
		HARVEY	HARVEY	BASE
			SHARON	NEW
			ED	NEW

In the example above the file named CHANGED had decks that also occurred on the file named BASE. These were selected rather than their duplicates from BASE, but were placed on NEWBASE in the same relative position as their counterparts. The decks from NEW which were not duplicated were added at the end of the deck sequence from BASE in the order they occurred on NEW. Likewise had there been any decks on CHANGED that had unique names they would have been added at the end of the deck sequence on NEWBASE.

In the example above files CHANGED and NEW had no decks by the same name. Had deck BOB also occurred on NEW, the copy of BOB from CHANGED would still have been selected and placed in the same position on NEWBASE. Had deck SHARON also occurred on CHANGED the copy from CHANGED would have been selected and it would have been placed in the same position as shown in the example.

146 48 48

3-8

1

5

7

31

32

33

34

35 36

37

138

139

140

141

142

143

146

			•		Version)	REV: J	
3.0	COMI	MANDS				a a a a a a a a a a a a a a a a a a a	

### 3.1.5 ADD_LIBRARY : ADL

3.1.5 ADD_LIBRARY | ADL

This command merges the decks from a list of source libraries into a single result library. The result library will include one deck by every name encountered on any of the source libraries. If decks by the same name occur on more than one of the libraries these decks will not be written on the result library and the duplication will be reported in the listing. Decks will have the following logical order on the result library:

- 1. All decks occurring on the last named source library will occur first in their original order.
- 2. Next the decks occurring on the next to last named library will occur in the order they occurred on that library.
- Decks occurring on the remaining libraries will occur in turn in the order they occurred on those libraries.

All source libraries must have the same key character. The result source library will have the same authorities, description, etc. as the last named source library.

add_library base=<file_name> [result=<file_name>] [list=<local_file_name>] [brief 1 full] [status=r11r21r31ef]

> base I ba : This is a list of names of files containing source libraries to be combined. The names must be enclosed in parentheses and separated by commas.

result 1 r : This is the file to receive the combined source library. If this parameter is not given, the combined source library will be written to a file named RESULT.

list 1 1 : This is the name of the file which receives the listing. The list shows from which source library decks on the result file were taken. If this parameter is not given, the file name OUTPUT will be used.

brief 1 br 1 full 1 fu 1 These keywords can be used to 147 select a shorter or more detailed listing. Currently 148

1 1 2

1 3

1 5

: 5

1 6

1 7

: 8

9

110

112

113

115

116

117

119

120

121

123

12%

125

127

128

129

138

131 32

133

134

135

137

138

139

140

142

143

144

145

03/24/80

1 1 2

13

15

1 8

1 9

110

111

112

8

ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS

3.1.5 ADD_LIBRARY : ADL

both listings are the same.

status 1 st : See basic status concept.

Example: scu.add_library (green,new,base) newbase

In this example the decks on the libraries on the files named GREEN and NEW are added to those on the library on file base to form a result on a file named NEWBASE. To illustrate how decks are selected and ordered during the processing of this command let us assume the three libraries being combined each contained the following decks:

<u>GREEN</u> FRED	<u>NEW</u> Sharon	<u>BASE</u> JIM	<u>NEWBASE_</u> JIM	<u>Deck Taken From</u> BASE
GORDY	ED	BOB	BOB	BASE
AL	-	MIKE	MIKE	BASE
JUDY		BILL	BILL	BASE
		SHERMAN	SHERMAN	BASE
		CURT	CURT	BASE
		JOHN	JOHN	BASE
		HARVEY	HARVEY	BASE
			SHARON	NEW
			ED	NEW
			FRED	GREEN
			GORDY	GREEN
			AL	GREEN
			JUDY	GREEN

In the example above there were no deck names duplicated on the three libraries. On the file NEWBASE the decks from BASE occur first in their original order followed by the decks from NEW and GREEN.

Had any deck names been duplicated, none of the duplicates would have been written to NEWBASE and the user would have been informed.

> 38 39 48

> 30

13f

:32

133

134

135

136

137

4**1** 

42 43

- 45 46
- 47
- 48

## 3-11

112

1 3

1 6

1 5

: 6

1 7

1 8

: 9

:10

111

113

114

116

118

118

119

120

121

123

124

125

126

127

129

130

131

133

134

135

13.6

138

139

140

141 42

143

144

145

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J

### 3.0 COMMANDS

### 3.1.6 REPLACE_LIBRARY I REL

### 3.1.6 REPLACE_LIBRARY I REL

This command merges the decks from a list of source libraries into a single result library. The result library will include one deck by every name encountered on the last named source library. If decks by the same name occur on more than one library the deck from the library listed first will be selected. Decks with names that do not occur on the last named library will not be included in the result library and that fact will be reflected in the listing. Decks will have the following logical order on the result library:

1. All decks with names occurring on the last named source library will appear in their original order.

All source libraries must have the same key character. The result source library will have the same authorities, description, etc. as the last named source library. This command will be indicate an error if there are any interlock violations (see discussion of interlock under EXTRACT_LIBRARY).

replace_library base=<file_name> [result=<file_name>] [list=<local_file_name>] [brief 1 full] [status=r11r21r31ef]

> base 1 ba : This is a list of names of files containing source libraries to be combined. The names must be enclosed in parentheses and separated by commas.

> result 1 r : This is the file to receive the combined source library. If this parameter is not given, the combined source library will be written to a file named RESULT.

list 1 1 : This is the name of the file which receives the listing. The list shows from which source library decks on the result file were taken. If this parameter is not given, the file name OUTPUT will be used.

brief | br | full | full These keywords can be used to select a shorter or more detailed listing. Currently both listings are the same.

status 1 st 1 See basic status concept.

1 1

123

1 5

: 5

1 7

1 8

121

122

123

125

125

1 6

					03/24/80
				170 Version)	REV: J
~~~~~	 	 	~ ~ ~ ~		to any sup the star star and and any sup star for the star star star star star star star star

3.0 COMMANDS

3.1.6 REPLACE_LIBRARY 1 REL

Example: scu.replace_library (changed, new, base) newbase

.

In this example two source libraries on files named CHANGED and NEW were combined with one on a file called BASE to form a result on a file named NEWBASE. To illustrate how decks are selected and ordered during the processing of this command, let us assume that the three libraries being combined each contained the following decks:

CHANGEDCURT	NEW_ BILL	<u>BASE</u> JIM	NEWBASE_	<u>Deck Taken From</u> CHANGED
JIM	HARVEY	BCB	BOB	CHANGED
MIKE		MIKE	MIKE	CHANGED
808		BILL	BILL	NEW
		SHERMAN	SHERMAN	BASE
		CURT	CURT	CHANGED
		JOHN	JOHN	BASE
		HARVEY	HARVEY	NEW

In the example above the files named CHANGED and NEW had decks that also occurred on the file named BASE. These were selected rather than their duplicates from BASE, but were placed on NEWBASE in the same relative position as their counterparts.

Had there been decks on CHANGED or NEW that did not appear on BASE, they would not have been written to NEWBASE and the user would have been informed.

33344444

45 45 45

47 48

COMPANY PRIVATE

03/24	180
for Source Code Utility (NOS 170 Version) REV:J	-
COMMANDS	5
7 EXTRACT_LIBRARY 1 EXL	
3.1.7 EXTRACT_LIBRARY I EXL	
This command is used to isolate a source library with only	
subset of decks on it for manipulation by the SCU editor or	
aintenance as a separate source library. If the INTERLOCK	
ption is used, the user must have write access to the file	
amed with the BASE parameter.	
extract_library [name= <deck_name>tall]</deck_name>	
[criteria= <local_file_name>] [interlock=<user_name> no_interlock]</user_name></local_file_name>	
[base= <file_name>]</file_name>	
[result= <file_name>]</file_name>	
[status=r11r21r31ef]	
name 1 na 1 all : This is the name of a deck to be	
extracted. Alternatively a list of names separated by	
commas and enclosed by parentheses may be given or a range of decks may be indicated by giving the first and	
last deck names to extract separated by ellipses. Use of	
the alternate keyword ALL to specify this parameter has	
the same effect as specifying the first and last decks as	
a range. If this parameter is omitted, the contents of	
the source library to be extracted will be determined	t
solely by the contents of the criteria file.	
critoria for t This error a file which can be used to	
criteria : cr : This names a file which can be used to further define which source data is to be included on the	
subset source library.	5
interlock int no_interlock nint : This parameter	•
provides a means to allow only the people authorized to	
use a particular user name to alter the decks being	•
written to the subset library. The value of the	
parameter may be given as a user name or if the keyword INTERLOCK only is given the user name the command is	
executed under will be used. When a deck is interlocked.	
an attempt to extract it to a subset library setting the	
interlock will be rejected. Also an attempt to replace	
the deck on a source library by doing a COMBINE_LIBRARY	
or REPLACE_LIBRARY with the base will be rejected, unless	
its from the subset library with the interlock set. If	
the command includes the NO_INTERLOCK keyword the user will be allowed to extract a subset source library	
whether or not the decks he requests are interlocked. An	
attempt to interlock decks with this command will be	
rejected if the criteria file contains directives which	
cause modifications to be excluded. If neither keyword	

COMPANY PRIVATE

· · · ·

		03/24/80
(S TOP	<pre>Source Code Utility (NOS 170 Version)</pre>	REVIJ
	IMANDS EXTRACT_LIBRARY I EXL	
		ينيو الله فين علي الله الله الله الله الله الله الله ال
	is present, the command is processed as if had been specified.	NO_INTERLOCK
	base I ba I This is the name of the file base source library. If this parameter is attempt is made to access a file with the n	not given an
	result 1 r : This is the file to rece source library. If this parameter is no subset source library will be written t RESULT.	t given, the
	status 1 st = See basic status concept.	
	extract_library_dsd_int=johndoe_r=temppl	
scu.	Text act_TID aly usu Int_Jumidue - Tempor	
deck	dsd. This subset source library is writte	
deck name	k dsd. This subset source library is writte ed TEMPPL and the deck dsd is interlocked NDOE.	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file
deck name	ed TEMPPL and the deck dsd is interlocked	n to a file

LUC SUFTWARE ENGINEERING STATEM	03/24/80
RS for Source Code Utility (NOS 170 Version)	REV: J
.0 COMMANDS	
.2 SOURCE DECKS	
	dig na cu su hu na na cu su na da dip na na ra
3.2 SOURCE DECKS	
Jec Jourge Deorg	
These commands are available for creating and	manipulating
source decks.	
2 2 4 CREATE DECK 1 CRD	
3.2.1 CREATE_DECK I CRD	
This command is used to create source deck(s).	
<pre>create_deck name=((<deck_name>[,<local_file_name>]</local_file_name></deck_name></pre>),) 1
names_on_file= <local_file_name></local_file_name>	
modification= <modification_name> [after=<deck_name>1before=<deck_name>1</deck_name></deck_name></modification_name>	alababatict
[author= <author_name_string>]</author_name_string>	albuane (Ic)
[description= <deck_description_string></deck_description_string>	3
[processor = <processor_name_string>]</processor_name_string>	-
[group= <group_name>]</group_name>	
[tab= <tab_character>Inotab]</tab_character>	
[columns= <tab_column_integers>]</tab_column_integers>	
[width= <integer>] [lid=<line_identifier_placement_keywor< td=""><td>f ch</td></line_identifier_placement_keywor<></integer>	f ch
[base= <file_name>]</file_name>	
[result= <file_name>]</file_name>	
[expand no_expand]	
[multi_partition single_partition]	
[same_as= <deck_name>] [status=r1!r2!r3!ef]</deck_name>	
LSTalus-ritrzirsteri	
name 1 na 1 names_on_file 1 nof : This param	eter must be
given in one of the two forms shown below. I	
or NA keyword is used for no keyword is	
required portion of this parameter is a 1 to	
deck name. It may optionally be followed by	
a local file containing source data to be inc deck as text. Alternatively a list of deck n	
names and file names may be given (see th	
Lists section of the NOS/VE ERS). All decks	
a single CREATE_DECK command will have	Identical
descriptive information taken from the othe	
on this command. An attempt to create a dec	k with the

descriptive information taken from the other parameters 140 on this command. An attempt to create a deck with the 41 same name as one already on the base source library will 142 be rejected. If the keyword NAMES_ON_FILE or NOF is 143 used, this parameter names a file from which initial text 144 for decks will be read. Deck names will be taken from 145 *DECK text embedded directives intermixed with the 146 initial text.

- • •
- 48

	03/2	4/80
RS for	Source Code Utility (NOS 170 Version) REV:J	
.0 COM	IMANDS	
.2.1 (CREATE_DECK 1 CRD	~~~~
	modification 1 mod : This 1 to 9 character modification	n
	name will be associated with the creation of this dec	
	and included as part of the line identifier for al	
	initial lines of text if any are introduced with thi command.	S
	after 1 af 1 before 1 be 1 alphabetic 1 alpha : Thi	s
	parameter can be used to name a deck on the library afte	r.
	or before which this deck will be added. If the keyword	
	AFTER or BEFORE are used alone, the deck will be added a	
	the end or beginning of the the deck sequence	
	respectively. Alternatively the ALPHABETIC keyword ma	-
	be used to indicate that the deck will be added in alphabetic sequence. If this parameter is omitted th	
	deck will be added in alphabetic sequence.	e
	and atts of agges to dibudneling seducines.	
	author 1 au : This 1 to 31 character string is provide	d
	to identify the person who wrote this unit of source	
	data.	
	description 1 de : This 1 to 256 character string is use	d
	to describe this ceck.	U
	to describe this terns	
	processor p : This 1 to 31 character string is used to	0
	identify the processor that will use the source data i	n
	this deck as input.	
	group 1 g : This is a 1 to 31 character group name to b	9
	associated with this deck. Alternatively a list of grou	
	names may be given separated by commas and enclosed in	
	parentheses.	
	tobal notable. If the TAR knowland is used on this	-
	tab 1 notab : If the TAB keyword is used or this parameter is given positionally, this one character	
	string specifies a default tab character for use when	
	editing this deck. If the NOTAB keyword is given tabbin	
	will be deselected by default for this deck.	
	columns col : These integers between 1 and 256 are th	P
	default tab stops when editing this deck. A series o	
	tab columns may be specified as a value list (see th	
	Parameter lists section of the NOS/VE ERS). A maximum o	
	256 tab columns may be specified.	
	width I w : This is the default line width to be use	
	when this deck is expanded and to be used by the SC	
	editor. The value must be given as an integer between	
	and 256. Specifying 0 as a width selects the defaults o	L.

5 10r	Source Code Utility (NOS 170 Version) REV:J
O COMM	ANDS EATE_DECK 1 CRD
COL GR	EAIE_UEUN I UKU waxaa waxaa
	supplying no trailing blanks when this deck is expanded
	and allowing lines up to 256 characters to be entered
	with the editor. If this parameter is not given a
	default value of 0 will be used.
	lid # This parameter is the default line identifier
	placement when this deck is expanded. If this parameter
	is omitted the expanded text file will contain no line
	IdentIfIers. This parameter can be given as one of 3
	values.
	nink a line identifiane with he strend as the
	right : Line identifiers will be placed on the right.
	left # Line identifiers will be placed on the left.
	none : Output lines will contain expanded text
	only.
	base 1 ba : This is the name of the file containing the
	base source library to which the new decks are to be
	added. If this parameter is omitted an attempt is made
	to access a file named BASE.
	result 1 r : This is the file to receive the modified
	source library. If this parameter is not given, the modified source library will be written to a file named
	RESULT.
	expand 1 exp 1 no_expand 1 nexp : If the NO_EXPAND
	keyword is included on this command these decks will only
	be written to the expanded text file produced by the
	EXPAND_DECK command as a result of processing the text embedded directives *COPY and *COPYC. If the EXPAND
	embedded directives *COPY and *COPYC. If the EXPAND keyword is included or this parameter is omitted
	entirely, these decks will be written to the expanded
	text file by the EXPAND_DECK command whether named on the
	command or on *COPY and *COPYC directives.
	multi_partition mp single_partition sp = If the
	MULTI_PARTITION keyword is included on the command, each
	end of record encountered in the source file will be represented on the source library as a *WEOP text
	embedded directive. If the SINGLE_PARTITION keyword is
	used or this parameter is omitted entirely, only the text
	before the first separator on the source file will be
	represented on the source library.

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS 3.2.1 CREATE_DECK | CRD 1 same_as 1 sa : This specifies the name of a deck on the 2 base library. Values from that deck's header will be 3 to supply values not given explicitly on this 1 6 used command for the author, description, processor, group, 1 5 tab. columns, width, lid, and expand/no_expand 1 6 parameters. 1 7 8 status 1 st : See basic status concept. 1 9 18 Example 1: 111 112 scu.create_deck .. 113 name=((amp\$get_direct,srcfile)) .. 14 modification=original .. 15 author="R. D. Palm" ... 16 description="Get Direct module." ... 117 processor="CYBIL/CI" .. 118 group=file_io ... 19 same_as=file_lo_prototype 20 21 22 In this example, a new deck named amp\$get_direct is added 123 to the source library on file BASE. The enlarged library is 124 written on the file RESULT. The initial text for the deck is 125 read from file SRCFILE and each line will have the 26 The author, modification name original associated with it. 27 description, processor and group parameters were given 28 explicitly on the command. Values for the 29 remaining parameters will be copied from the description of the deck 30 named file_lo_prototype. In this example the ability to use 31 an ellipsis at the end of a line to continue a command was 32 used to make the command more readable and possibly easier to 33 maintain or modify should it be used as part of a procedure 34 file. 35 36 Example 2: 37 138 scu.crd William_Cody bill 139 140 In this example the deck WILLIAM_CODY is added to the 141 library on file BASE and the result is written to file 142 RESULT. The creation of the deck is associated with the :43 modification name bill. No initial text is supplied for the 44 deck (it may be added later with the editor) and default 45 values are supplied for all the other parameters. 46 47 Example 3: 48 COMPANY PRIVATE

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J

3.0 COMMANDS 3.2.1 CREATE_DECK | CRD

scu.create_deck ..
(u,v,(w,wfile),x,y,z) ..
encrypto ..
author="M. J. Perreten" ..
same_as=CYBIL_type_declarations

In this example decks u through z are added to the source library on file BASE. The enlarged library is written to the file RESULT. Deck w has initial text supplied from file WFILE. The modification and author fields are given explicitly on the command. Values for the remaining parameters will be copied from the description of the deck named CYBIL_type_declarations.

Example 4:

scu.create_deck ..
nof=august ..
number12 ..
author="Henry McGilton" ..
description="August Tools Bulletin"

In this example text for some deck or decks is read from a file named August which contains DECK text embedded directives. This deck will be added to those on the library on file BASE and the result written to a file named RESULT. The modification name associated with each line in the new deck will be NUMBER12. Henry McGilton is identified as the author and the deck is described as containing the August Tools Bulletin.

:30

1 5

SOF	TWARE ENGINEERING SYSTEM 3-20
	03/24/0
5 for	Source Code Utility (NOS 170 Version) REV:J
	MANDS
2.2 0	ISPLAY_DECK DID
3.2.	2 DISPLAY_DECK DID
т	his command displays descriptive information about the
deck	•
disp	lay_deck_name= <deck_name>tall</deck_name>
	[base= <file_name>]</file_name>
	[list= <local_file_name>]</local_file_name>
	[brlef 1 full]
	[notext] active] inactive] [status=r11r21r31ef]
	[518105-11112113161]
	name 1 na 1 all # This is the name of a deck to be
	displayed. Alternatively a list of names separated by
	commas and enclosed by parentheses may be given or a
	range of decks may be indicated by giving the first and
	last deck names to display separated by ellipses. Use of
	the alternate keyword ALL to specify this parameter has the same effect as specifying the first and last decks as
	a range.
	have I be I This is the same of the file containing the
	base I ba # This is the name of the file containing the base source library. If this parameter is not given an
	attempt is made to access a file with the name BASE.
	list 1 1 : This is the name of the file which receives
	the listing. If this parameter is not given, the file
	name OUTPUT will be used.
	brief t br t full t fu : If the BRIEF keyword is included
	on this command, only the descriptive information from
	the deck header will be displayed. If the FULL keyword
	is used the list of modification names and group names
	associated with this deck will be displayed as well.
	Omitting this parameter gives the brief display.
	notext 1 active 1 act 1 inactive 1 inact: If the
	INACTIVE keyword is included on this command both active
	and inactive lines from the deck will be displayed. If
	the keyword ACTIVE is given only the active lines in the
	deck will be displayed. If the NO_TEXT keyword is given
	or this parameter is omitted entirely none of the text
	lines of the deck will be displayed.
	status 1 st : See basic status concept.
	STATUS I ST + SEE DASTC STATUS CONCEPT.
Exam	ple:

3-21 03/24/80

ERS for Source Code Utility (NOS 170 Version) REV:J

3.0 COMMANDS

3.2.2 DISPLAY_DECK | DID

The command in this example lists the descriptive 1 2 information from the header of deck dsd from the source 1 3 library on file BASE as well as the list of modification and 1 5 group names associated with this deck. The text lines of the 1 5 deck are not included in the listing. The listing is written 1 6 to the file called OUTPUT. 1 7

DC SOFTWARE ENGINEERING SYSTEM RS for Source Code Utility (NOS 170 Version) REV:	03/24/80 J
•0 COMMANDS	19 19 19 19 19 19 19 19 19 19 19 19 19 1
2-3 CHANGE DECK : CHD	
	یک بیاد دیاد بیاد میلدوند ویک داد د
3.2.3 CHANGE_DECK I CHD	
This command is used to change fields in the description.	deck
change_deck_name= <deck_name>lall</deck_name>	
[author= <author_name_string>]</author_name_string>	
[Interlock= <user_name>]</user_name>	
[description= <deck_description_string>]</deck_description_string>	
[processor= <processor_name_string>] [group=<group_name>]</group_name></processor_name_string>	
[delete_group= <group_name>]</group_name>	
[tab= <tab_character>inotab]</tab_character>	
[columns= <tab_column_integers>]</tab_column_integers>	
[delete_columns= <tab_column_integers>]</tab_column_integers>	
[width= <in teger="">]</in>	
[lid= <line_identifier_placement_keyword>]</line_identifier_placement_keyword>	
[base= <file_name>] [result=<file_name>]</file_name></file_name>	
[expand 1 no_expand]	
[status=r11r21r31ef]	
name I na I all : This is the name of a deck to hav	
description altered. Alternatively a list of	
separated by commas and enclosed by parentheses r	•
given or a range of decks may be indicated by giving first and last deck names separated by ellipses.	
the alternate keyword ALL to specify this parameter	
the same effect as specifying the first and last de	
a range.	
author I au I This 1 to 31 character string is pro	
to identify the person who wrote this unit of :	source
data.	
interlock 1 int : This parameter is included to all	nw the
interlock for a deck (see the EXTRACT_LIBRARY comman	
be changed or cleared. It is altered by supplying a	
user name as the value for this parameter and	
cleared by supplying the keyword INTERLOCK	only.
Altering this field is restricted to the user name	
has authority 4 for this library. Clearing this fid	
restricted to the user name which has authority 4 for library or the user name for which the interlock is	
ITALA A THE OPEN HOUSE FOR MUTCH THE TURNING 10CK 12	3416
description 1 de : This 1 to 256 character string is	s use d
to describe the contents of this deck.	

ERS for	03/24/8 Source Code Utility (NOS 170 Version) REV:J	s 0
3.0 COM 3.2.3 C	IMANDS CHANGE_DECK 1 CHD	
	processor 1 p : This 1 to 31 character string is used to identify the processor that will use the source data in this deck as input.	
	group 1 g : This is a 1 to 31 character group name to, be associated with this deck. Aternatively a list of group names may be given separated by commas and enclosed in parentheses.	
	delete_group 1 dg # This parameter allows group names to be deleted from the deck header. The value of this parameter is given as a single group name or a list of group names separated by commas and enclosed in parentheses.	
	tab I notab # If the TAB keyword is used or this parameter is given positionally, this one character string specifies a default tab character for use when editing this deck. If the NOTAB keyword is given tabbing will be deselected by default for this deck.	
	columns 1 col : These integers between 1 and 256 are tab stops to be added to the deck header. A series of tab columns to be added may be specified as a value list (see the Parameter lists section of the NOS/VE ERS). A maximum of 256 tab columns may be specified. When the delete_columns parameter is also included on this command, the tab stops it specifies are cleared before the tab stops specified by the columns parameter are set.	
	delete_columns dcol : These integers between 1 and 256 are default tab stops for editing to be deleted from the deck header. A series of tabs stops may be specified as a value list (see the Parameter lists section of the NOS/VE ERS).	
	width 1 w : This is the default line width to be used when this deck is expanded and to be used by the SCU editor. The value must be given as an integer between 0 and 256. Specifying 0 as a width selects the defaults of supplying no trailing blanks when this deck is expanded and allowing lines up to 256 characters to be entered with the editor.	-1 ¹
	lid : This parameter is the default line identifier placement when this deck is expanded. This parameter can be given as one of 3 values.	

for	Sourc	e Code	Utilit	Y (NOS 17	0 Versi	on)	REV:	03/24/80 J
		DECK 1	СНД		~~~~~~~~			
	~~~~~				~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	au au au au au au au au au a		
				e identi	fiers	will be	placed o	on the
		left :	Line i	dentifier	s will	be placed	d on the	left.
			: Outpu	ut lines	WITT	contain	expanded	text
	base	sourc	e libra	ary. If	this pa	rameter i	is not giv	ien an
	sourc modif	e lib Ied so	rary.	If this	param	eter is	not giver	, the
	keywo be w EXPAN embed keywo entir text	rd is ritten D_DECK ded d rd is ely, file b	include to th comman irective includ these y the E	d on this ne expan d as a re es *COPY ded or decks wi XPAND_DEC	comman ded te sult of and this p 11 be K comma	d these ( xt file ; process *COPYC. arameter written t nd whethe	decks will broduced t sing the If the E is on to the exp	only by the text EXPAND aitted banded
	statu	s t st	: See I	basic sta	tus con	cept.		
		m\$dele	te p="C	YBIL"				
will SCM\$	use t delete	his de Will	ck as in be of	nput. T btained f	he des rom fil	cription e BASE ar	of the nd the mod	deck
		COMMANDS 3 CHANGE_ 3 CHANGE_ 3 CHANGE_ base attem resul sourc modif RESUL expan keywo be w EXPAN embed keywo entir text comma statu Example: SCU.CHD sc In thi will use t scm\$delete	COMMANDS 3 CHANGE_DECK 1 right right. 1 eft : none only. base 1 ba : base sourc attempt is result 1 r source 1 ib modified so RESULT. expand 1 keyword is be written EXPAND_DECK embedded d keyword is entirely, text file b command or status 1 st Example: SCU. CHD scm\$dele In this example scm\$delete will	COMMANDS 3 CHANGE_DECK : CHD right : Line right. left : Line id none : Output only. base ! ba : This is base source libra attempt is made to result ! r : This source library. modified source lil RESULT. expand ! exp ! keyword is included be written to th EXPAND_DECK command embedded directive keyword is included be written to th EXPAND_DECK command embedded directive keyword is included be tirely, these of text file by the Ei command or on *COPY status ! st : See 1 Example: SCU.CHD scm\$delete p=*CY will use this deck as in scm\$delete will be of	COMMANDS 3 CHANGE_DECK 1 CHD right : Line identifier none : Output lines only. base 1 ba : This is the nam base source library. If attempt is made to access a result : r : This is the fi source library. If this modified source library wil RESULT. expand 1 exp 1 no_expan keyword is included on this be written to the expan EXPAND_DECK command as a re embedded directives *COPY keyword is included or entirely, these decks wi text file by the EXPAND_DEC command or on *COPY and *CO status : st : See basic sta Example: SCU.CHD scm\$delete p=*CYBIL's In this example CYBIL is id will use this deck as input. T	COMMANDS 3 CHANGE_DECK : CHD right : Line identifiers right. left : Line identifiers will none : Output lines will only. base ! ba : This is the name of th base source library. If this pa attempt is made to access a file w result : r : This is the file to source library. If this param modified source library will be wr RESULT. expand 1 exp 1 no_expand 1 n keyword is included on this comman be written to the expanded te EXPAND_DECK command as a result of embedded directives *COPY and keyword is included or this p entirely, these decks will be text file by the EXPAND_DECK comman command or on *COPY and *COPYC dir status ! st : See basic status con Example: SCU.CHD scm\$delete p=*CYBIL* In this example CYBIL is identifie will use this deck as input. The des scm\$delete will be obtained from fil	COMMANDS 3 CHANGE_DECK : CHD right : Line identifiers will be right. left : Line identifiers will be placed none : Output lines will contain only. base ! ba : This is the name of the file of base source library. If this parameter is attempt is made to access a file with the r result ! r : This is the file to receive source library. If this parameter is modified source library will be written to RESULT. expand ! exp ! no_expand ! nexp : If keyword is included on this command these be written to the expanded text file of EXPAND_DECK command as a result of process embedded directives *COPY and *COPYC. keyword is included or this parameter entirely, these decks will be written to text file by the EXPAND_DECK command whether command or on *COPY and *COPYC directives. status ! st : See basic status concept. Example: SCU.CHD scm\$delete p=*CYBIL* In this example CYBIL is identified as the will use this deck as input. The description scm\$delete will be obtained from file BASE ar	COMMANDS 3 CHANGE_DECK 1 CHD right : Line identifiers will be placed of right. left : Line identifiers will be placed on the none : Output lines will contain expanded only. base 1 ba : This is the name of the file containing base source library. If this parameter is not give result : r : This is the file to receive the mod source library. If this parameter is not giver modified source library will be written to a file RESULT. expand 1 exp 1 no_expand 1 nexp : If the NO_E keyword is included on this command these decks will be written to the expanded text file produced the EXPAND_DECK command as a result of processing the embedded directives *COPY and *COPYC. If the EXPAND_DECK command whether named of command or on *COPY and *COPYC directives. status i st : See basic status concept. Example:

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS 3.2.4 PURGE_DECK 1 PUD 3.2.4 PURGE_DECK 1 PUD 1 2 This command is used to remove decks from a source 3 library. This command will be rejected if the user name it is 毛 executed under does not have an authority equal to or higher 5 than the state of the modification associated with this deck*s 6 creation. 7 8 purge_deck_name=<deck_name> 9 [base=<file_name>] :10 [result=<file_name>] 111 [status=r11r21r31ef] 112 13 name I na # This is the name of a deck to be purged. 14 Alternatively a list of names separated by commas and 15 enclosed by parentheses may be given or a range of decks 16 may be indicated by giving the first and last deck names 17 to purge separated by ellipses. 18 19 base 1 ba # This is the name of the file containing the 120 base source library. If this parameter is not given an 121 attempt is made to access a file with the name BASE. 122 23 result 1 r : This is the file to receive the modified 124 source library. If this parameter is not given, the 125 modified source library will be written to a file named 126 **RESULT**. 127 28 status 1 st : See basic status concept. 129 30 Example: 131 scu.purge_deck soup..nuts current shorter 132 38 In this example the source library on file SHORTER will 34 contain all of the decks in the source library on file CURRENT 35 except those in the range from soup to nuts. 36 37 38 39 40 4I 42 43 45 45 46 47 48

12

3

ъ

5

6

7

8

9

10

11 12

13

14

115

116

117

19

20

21

22

23

25

25

26 27

128

129

130 31

132

					03/24/80
			-	Version)	REV: J
	. an an an an an	 *****	~~~~~		ng dag sing dag sing mar mar pag dag dag sing sing dag dag sing dag dag dag dag dag dag dag dag dag da

### 3.0 COMMANDS

### 3.2.5 SEQUENCE_DECK 1 SED

### 3.2.5 SEQUENCE_DECK | SED

This command is used to resequence a deck. Only lines associated with modifications of state 4 will be altered. The modification identifier under which the deck was created will be associated with each of these lines on the result library. All history information concerning previous modifications to these lines will be discarded, including inactive lines. Because of this, resequencing is recommended only when a deck has attained some significant milestone. This command is rejected if it is not executed under the user name which has authority 4 for the library.

# sequence_deck name=<deck_name>lall [base=<file_name>] [result=<file_name>] [status=r11r21r31ef]

name I na I all : This is the name of a deck to be resequenced. Alternatively a list of names separated by commas and enclosed by parentheses may be given or a range of decks may be indicated by giving the first and last deck names to resequence separated by ellipses. Use of the alternate keyword ALL to specify this parameter has the same effect as specifying the first and last decks as a range.

base 1 ba : This is the name of the file containing the base source library. If this parameter is not given an attempt is made to access a file with the name BASE.

result 1 r : This is the file to receive the modified source library. If this parameter is not given, the modified source library will be written to a file named RESULT.

status 1 st 1 See basic status concept.

Example: scu.sequence_deck_name=(Bob,Carol,Ted,Alice,a..z)_oldpl,r=joe

139

> 45 46 47

COMMANDS .6 MOVE_DECK 1 MOD	****
3.2.6 MOVE_DECK # MOD	
This command is used to move a deck to anoth position on the source library.	her logical
move_deck_name= <deck_name>tall</deck_name>	
[after= <deck_name>1before=<deck_name>1a1; [base=<file_name>]</file_name></deck_name></deck_name>	phabet ic i
[result= <fite_name>]</fite_name>	
[status=r11r21r31ef]	
name I na I all : This is the name of a	
moved. Alternatively a l'1st of names separate	
and enclosed by parentheses may be given or decks may be indicated by giving the first and	-
names to move separated by ellipses. An er	
reported if an attempt is made to move a range	
to a destination within the range. The	
keyword ALL will be allowed only if alphabeti	ic ordering
is selected and will have the effect of sortir on the result library.	ng the decks
after 1 af 1 before 1 be 1 alphabetic 1 alp	ha : This
parameter can be used to name a deck on the 11	
or before which this deck will be moved. If 1	
AFTER or BEFORE are used alone, this deck wi	
to the end or beginning of the deci	* -
respectively. Alternatively the ALPHABETIC	•
be used to indicate that the deck will be alphabetic sequence. If this parameter is	e moved to
deck will be moved to alphabetic sequence.	omitted the
base I ba : This is the name of the file cont	aining the
base source library. If this parameter is r attempt is made to access a file with the name	
result t r : This is the file to receive th	
source library. If this parameter is not	
modified source library will be written to a RESULT.	file named
status 1 st : See basic status concept.	
Example:	
scu.move_deck r=newp1 na=Anatoly after=Bobby oldp1	
In this example the use of keywords has	allowed the
	ALIOWED THE

### 03/24/80

ERS	for	Source	Code	Utility	(NOS	170	Version)	REV	:J
** ** **	***						ورد ماه برد جار جار ها مه به ور جار ج		

### 3.0 COMMANDS 3.2.6 MOVE DECK 1 MOD

 هذه مدر	

sequence.

### COMPANY PRIVATE

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS 3.2.7 EXPAND DECK 1 EXPD 3.2.7 EXPAND_DECK 1 EXPD t 2 This command expands a source deck into a file where it may 3 be used as input to some processor. 4 During the course of 5 expansion, text embedded directives intermixed with the source 6 will be processed. For a description of these, see the 7 section on text embedded directives. 8 9 expand_deck [name=<deck_name>1a11] [compile=<local_file_name>] 10 [criteria=<local_file_name>] 11 [width=<integer>] 12 [lid=<line_identifier_placement_keyword>] 13 [base=<file_name>] 114 [list=<local_file_name>] 115 [depth=<integer>] 16 [brief!full] 117 [library_order | command_order] 18 [status=r11r21r31ef] :19 28 1 na 1 all : This is the name of a deck to be 21 name expanded. Alternatively a list of names separated by 22 commas and enclosed by parentheses may be given or a 23 range of decks may be indicated by giving the first and 24 last deck names to expand separated by ellipses. The use 25 of deck ranges on this command for anything other than 26 127 decks occurring on the last named source library is not recommended (See description of the LIBRARY ORDER keyword 128 below). Use of the alternate keyword ALL to specify this 29 31 parameter has the same effect as specifying all decks from a combined library as a range. If this parameter is, 31 omitted, the source data to be expanded 32 WITT he determined solely by the contents of the criteria file. 33 34 compile 1 c : This is the name of the file for the 135 expanded text. This file will be rewound before and 136 If this parameter is not after the command is processed. 137 given, the expanded text will be written to a file named 38 COMPILE. 39 48 criteria | cr : This names a file which can be used to 41 further define which source data is to be expanded. If 42 no criteria file is used, the selected decks will be 43 expanded with all modifications included. 44 45 width I w # This is the length of 46 the output lines excluding line identifiers if they are requested. 47 Δ maximum value of 256 will be allowed. An additional 17 48

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J ------3.0 COMMANDS 3.2.7 EXPAND_DECK | EXPD columns will be necessary for a line identifier since a Ť blank will separate the modification identifier from the 2 sequence number and another blank will separate the 3 entire line identifier from the expanded text. If 0 is 4 specified as the value for this parameter, output lines 5 will be actual length. If this parameter is not given on 1 6 the command, the default value from the deck header will 1 7 be used. If the width selected results in a line being 1 8 truncated, an error will be reported. : 9 10 lid : This parameter can be given as one of 3 values. 111 12 right : Line identifiers will be placed on the 113 right. 115 15 left : Line identifiers will be placed on the left. 116 17 none : Output lines will contain expanded text 118 only. 119 28 If this parameter is not given on the command, the 121 default value from the deck header is used. 122 23 base I ba ; This is the name of the file containing the 24 base source library. Alternatively a list of file names 25 may be given to allow decks to be expanded from alternate 126 source libraries. See normal search order. If this 127 parameter is not given an attempt is made to access a 128 file with the name BASE. 129 30 list 1 1 : This is the name of the file to receive the 131 listing. If this parameter is not given, the file name 132 OUTPUT will be used. 133 34 depth : This non-negative integer limits how deeply 135 nested *COPY or *COPYC text embedded directives will be 136 processed. *COPY or *COPYC directives encountered at the 137 next level will be included in the expanded text file as 138 text lines. If this parameter is omitted, *COPY on 139 *COPYC directives will be processed wherever they are 140 encountered. 41 42 brief 1 br 1 full 1 fu 1 If the BRIEF keyword is present 143 or this parameter is omitted, the list will contain only 144 a report of errors which may have occurred while 145 processing the command. If the FULL keyword is present, 146 an explanation of which library each deck was selected 47 from will be included when the command accesses multiple 148 COMPANY PRÍVATE

3-31

07/01/00

								03/24/00
ERS	for	Source	Code	Utility	(NOS	170	Version)	REV:J
***		د وند مد بد مدمد مد مد ب						ên de ha ha da da da wa ku

#### 3.0 COMMANDS

3.2.7 EXPAND_DECK | EXPD

source libraries.

library_order 1 to 1 command_order 1 co 1 If this command includes the COMMAND_ORDER keyword the decks will be written to the excanded text file in the order they are specified on the command. If the LIBRARY_ORDER keyword is included or this parameter is omitted, the decks are written' to the expanded text file in the order that the decks occur on the base source library (or in the order they would occur on a combined source library if one were produced from the list of libraries specified by the base parameter).

status 1 st : See basic status concept.

Example 1: scu.expand_deck mydeck ba=oldpl

In this example the deck mydeck is expanded from the source library on the file OLDPL and written on the expanded text file named COMPILE.

Example 2: scu.expand_deck scm\$change ba=(myopl,(cybccmn,ses))

In this example the deck scm\$change is expanded from the libraries MYOPL and CYCCCMN and written on the expanded text file named COMPILE. In this case CYBCCMN was located under another user name.

33333344444444

112

1 3

1 4

5

6

7

1

9

10

11 12

13

# 13/24/81

			00/24/00
ERS for Source	Code Utility	(NOS 170 Version)	REV: J
و جام		وی دوله دوله خود دید. ویک دوله دوله دوله دوله دوله دوله دوله دوله	د این کار محمد بیان کی باری کی بیان میں کی باری کی باری کی بیان کی کار

### 3.0 COMMANDS

3.2.8 EXTRACT_DECK : EXTD

3.2.8 EXTRACT_DECK 1 EXTD

This command performs a function similar to EXPAND_DECK with the exception that text embedded directives are not processed but included as text.

extract_deck [name=<deck_name>tall]
 [source=<local_file_name>]
 [criteria=<tocal_file_name>]
 [width=<integer>]
 [tid=<tine_identifier_placement_keyword>]
 [base=<file_name>]
 [list=<local_file_name>]
 [brieftfull]
 [library_order1command_order]
 [expandino_expand]
 [deck_directivestno_deck_directives]
 [status=r11r21r31ef]

name 1 na 1 all : This is the name of a deck to be extracted. Alternatively a list of names separated by commas and enclosed by parentheses may be given or a range of decks may be indicated by giving the first and last deck names to extract separated by ellipses. The use of deck ranges on this command for anything other than decks occurring on the last named source library is not recommended (See description of the LIBRARY_ORDER keyword below). Use of the alternate keyword ALL to specify this parameter has the same effect as specifying all decks from a combined library as a range. If this parameter is omitted, the source data to be extracted will be determined sclely by the contents of the criteria file.

source 1 s : This is the name of the file for the extracted text. Decks will be separated by end of partition (end of record on NOS170). This file will be rewound before and after the command is processed. If this parameter is not given, the extracted text will be written to a file named SOURCE.

criteria 1 cr : This names a file which can be used to further define which source data is to be extracted. If no criteria file is used, the selected decks will be extracted with all modifications included.

width 1 w = This is the length of the output lines 47 excluding line identifiers if they are requested. A 48

1 2

3

4

5 6

7

8

9

18

11

:12

113

114

15

117

118 19

120

121

122

128

124

125

126

127

28

29

38

31

32

33 34

135

:36

137

138

39

40 41

42

43

44

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS 3.2.8 EXTRACT DECK : EXTD maximum value of 256 will be allowed. An additional 17 1 columns will be necessary for a line identifier since a 2 blank, will separate the modification identifier from the 3 sequence number and another blank will separate the 6 entire line identifier from the extracted text. If 0 is 5 specified as the value of this parameter, output lines 6 will be actual length. If this parameter is not given on : 7 the command, the default value from the deck header will 8 be used. If the width selected results in a line being 1 9 truncated, an error will be reported. 110 11 lid : This parameter can be given as one of 3 values. 112 13 right : Line identifiers will be placed on the 14 right. 15 16 left : Line identifiers will be placed on the left. 17 18 19 none: Output lines will contain extracted text only. 20 2£ If this parameter is not given on the command, the 122 default value from the deck header is used. 123 24 base 1 ba : This is the name of the file containing the 25 base source library. Alternatively a list of file names 26 may be given to allow decks to be extracted from 127 alternate source libraries. See normal search order. If 128 this parameter is not given an attempt is made to access 129 a file with the name BASE. :30 3Ť list 1 1 : This is the name of the file to receive the 132 listing. If this parameter is not given, the file name 138 OUTPUT will be used. 134 35 brief 1 br 1 full 1 fu : If the BRIEF keyword is present 136 or this parameter is omitted, the list will contain only 137 report of errors which may have occurred while :38 а processing the command. If the FULL keyword is present, :39 an explanation of which library each deck was selected 140 from will be included when the command accesses multiple 141 source libraries. :42 43 library_order 1 to 1 command_order 1 co : If this command 144 includes the COMMAND_ORDER keyword the decks will be 145 written to the extracted text file in the order they are 46 specified on the command. If the LIBRARY_ORDER keyword 47 is included or this parameter is omitted, the decks are 48

Ť

2

3

4 5

6

: 7

1 8

: 9

110

111

112

113

114

116

117

:18

119

120

121

123 24 125

126

28

29

30

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J

#### 3.0 COMMANDS

3.2.8 EXTRACT_DECK ! EXTD

written to the extracted text file in the order that the decks occur on the base source library (or in the order they would occur on a combined source library if one were produced from the list of libraries specified by the base parameter).

expand 1 exp 1 no_expand 1 nexp = If the EXPAND keyword is present on this command, only decks without the NO_EXPAND attribute will be written to the SOURCE file. If the NO_EXPAND keyword is present, only decks with the NO_EXPAND attribute will be written to the SOURCE file. If neither keyword is present, decks will be written to the SOURCE file without regard to their NO_EXPAND attribute.

deck_directives 1 dd 1 no_deck_directives 1 ndd : If the DECK_DIRECTIVES k (yword is present, DECK text embedded directives will precede each deck on the SOURCE file. If the NO_DECK_DIRECTIVES keyword is used or neither keyword is present, no DECK directives will be written to the SOURCE file.

status 1 st : See basic status concept.

Example: scu.extract_deck_mydeck_ba=oldpl

In this example the deck mydeck is extracted from the source library on the file OLDPL and written on the text file named SOURCE.

#### 44 45

- 46
- 48 48

1 2

: 3

1 4

1 5

1 6

178

110

11**1** 

112

113

15

16

17

18

19

2021

22

23

:24

125

126 27 28

29

30 31

32

133

134

:35

136

13₹

138 39

148

142

148

9

							03/24	/80
ERS for	Source	Code	Utility	(NOS	170	Version)	REV: J	
~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~					ی براد جاد ماد براه مرد براه جاد ماد جاد جاد ماد ماد جاد جا		****
3.0 COM	MANDS							
_								

3.2.9 DISPLAY_DECK_REFERENCES | DDR

3.2.9 DISPLAY_DECK_REFERENCES | DDR

This command displays the names of the decks which reference the named deck directly or indirectly by *COPY text embedded directives. It can optionally also display the names of the decks referenced by the named deck directly or indirectly through use of the *COPY text embedded directive.

display_deck_references name=<deck_name>tall [base=<file_name>] [list=<local_file_name>] [brief!full] [status=r1!r2!r3!ef]

name 1 na 1 all : This is the name of a deck to be cross referenced. Alternatively a list of names separated by commas and enclosed by parentheses may be given or a range of decks may be indicated by giving the first and last deck names to cross reference separated by ellipses. Use of the alternate keyword ALL to specify this parameter has the same effect as specifying the first and last decks as a range.

base 1 ba : This is the name of the file containing the base source library. If this parameter is not given an attempt is made to access a file with the name BASE.

list 1 1 : This is the name of the file which receives the listing. If this parameter is not given, the file name OUTPUT will be used.

brief 1 br 1 full 1 fu : If the BRIEF keyword is included on this command or this parameter is omitted entirely, only the decks which reference this deck will be listed. If the FULL keyword is used, the list of decks referenced by this deck will be shown as well. References to decks not on the base source library will be flagged as external.

status 1 st : See basic status concept.

Example: SCU.DDR scc\$display_default_values scupt

The command in this example displays the names of all the 45 decks on the source library on file SCUPL which refer to the 46 deck scc\$display_default_values directly or indirectly through 147 *COPY directives. 148 CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J ------3.0 COMMANDS 3.2.10 DISPLAY_DECK_LIST | DDL 3.2.10 DISPLAY_DECK_LIST | DDL 1 2 This command displays the list of decks on a source 3 library. 4 5 display_deck_list [base=<file_name>] 1 6 [list=<local_file_name>] 1 7 [brief | full] 1 8 [status=r11r21r31ef] 1 9 10 base 1 ba : This is the name of the file containing the 11 base source library. Alternatively a list of file names 112 may be given to give a combined deck list. If this 113 parameter is not given an attempt is made to access a 114 file with the name BASE. 115 16 list 1 1 : This is the name of the file which receives 17 the listing. If this parameter is not given, the file 18 name OUTPUT will be used. 19 20 brief 1 br 1 full 1 fu : If the command contains the key 21 word BRIEF or this parameter is omitted, a shorter form 22 of the listing will be produced. If the FULL keyword is 23 present a more detailed listing will be given. Currently 24 these listings are identical. 25 26 status 1 st : See basic status concept. 27 28 Example: 129 SCU.DDL scupt sculist :30 31 32 33 35 35 36 37 38 39 40 41 42 43 44 45 46 47 48 COMPANY PRIVATE

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS 3.2.11 DISPLAY_GROUP 1 DIG 3.2.11 DISPLAY_GROUP | DIG 1 2 1 8 This command will display a list of the decks associated with a group and optionally the short form of the deck 1 4 1 5 description (see DISPLAY DECK) for all decks associated with this group name. 6 7 8 display_group group=<group_name> 1 9 [base=<file name>] [list=<local_file_name>] 110 [brief ! full] 111 [status=r11r21r31ef] 112 13 group 1 g : This is the name of the group. 14 15 base ; ba : This is the name of the file containing the 16 base source library. Alternatively a list of file names 117 may be given to allow a user to interrogate a set of 118 source libraries that are frequently used together. If 119 this parameter is not given an attempt is made to access 120 a file with the name BASE. 121 22 list 1 1 : This is the name of the file which receives 123 the listing. If this parameter is not given, the file 125 name OUTPUT will be used. 125 26 brief 1 br 1 full 1 fu : If the command includes the 127 keyword BRIEF, only the list of decks associated with the 128 group will be displayed. If the keyword FULL is present, 129 the short form of the deck description for each deck will 130 be given as well. The shorter display is selected by 131 default. 132 33 status 1 st : See basic status concept. 134 35 36 Example: :37 138 scu.display_group all_iou_programs oldpl 39 48 3.2.12 DISPLAY_GROUP_LIST | DGL 41 42 This command will list all the group names associated with 43 this source library as well as the decks associated with each 44 45 group. 46 display_group_list [base=<file_name>] 147 [list=<local_file_name>] 148

07/04/00

		U 3/24/8 U
ERS for	Source Code Utility (NOS 170 Version)	REVIJ
3.0 COMM	ANDS	
	ISPLAY_GROUP_LIST DGL	
14 44 44 44 44 44 44 44 44 44		in the district of the second seco
	[brief : full]	
	[status=r11r21r31ef]	· · · .
	base I ba : This is the name of the file conta	
	base source library. Alternatively a list of may be given to produce a combined group list.	
	parameter is not given an attempt is made	
	file with the name BASE.	
	died d. d. This is the same of the file which	
	list : This is the name of the file which the listing. If this parameter is not given	
	name OUTPUT will be used.	······································
	hated that first the the assess to	tudan Aba
	brief 1 br 1 full 1 fu : If the command in BRIEF keyword or this parameter is omitted en	
	display will contain only the group names. If	
	keyword is given, the group names as well as	s the decks
	associated with each group will be displayed.	
	status 1 st 1 See basic status concept.	
	······································	
Examp		
SCUAL	GL testpl	
	t.	

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS **3.3 MODIFICATIONS** 3.3 MODIFICATIONS 1 2 The following commands together with the SCU editor are 3 used to manipulate modifications to source libraries. 6 5 6 3.3.1 CREATE_MODIFICATION 1 CRM 7 8 9 be used to create the descriptive This command can information to be associated with a modification. 18 Modifications are always created with state 0. 11 12 create modification modification=<modification_name> 13 14 [feature=<feature_name>] [author=<author_name_string>] 15 [description=<descriptive_string>] 116 [base=<file_name>] 117 [result=<file_name>] 118 [status=r11r21r31ef] 119 28 modification 1 mod : This 1 to 9 character field is the 2 f modification name. Alternatively a list of names may be 122 given. 123 24 feature 1 fe : This is a 1 to 31 character feature name 125 to be associated with this modification. 126 23 author 1 au : This 1 to 31 character string can be used 128 to identify the individual who wrote this modification. 129 30 description 1 de : This 1 to 256 character string can be 13± used to describe the modification. It is suggested that 132 this be used to point to official documentation for the 133 134 modification such as psr numbers or design documents. 35 base I ba : This is the name of the file containing the 135 base source library. If this parameter is not given an 137 attempt is made to access a file with the name BASE. 138 39 result 1 r : This is the file to receive the modified 140 source library. If this parameter is not given, the 141 modified source library will be written to a file named :42 RESULT. 143 45 status 1 st : See basic status concept. 145 46 Example: 147 SCU.CRM .. 148

1

1 3

4 5

578

9 18

11

12 13

115

115

116

117

118

119

12T

122

:23

126

125

126

127

29

130

:31

32

33 35

135

136

137

39

48

41 42

: 2

ERS for Source Code Utility (NOS 170 Version) REV:J

3.0 COMMANDS

3.3.1 CREATE_MODIFICATION 1 CRM

mod=mymod ..
fe=my_appearance ..
au="M. J. Perreten" ..
de="short , fat, brown hair, beard" ..
ba=oldfact ..
r=newfact

3.3.2 DISPLAY_MODIFICATION 1 DIM

This command displays descriptive information about a modification.

display_modification modification=<modification_name>tall [name=<deck_name>] [base=<file_name>] [list=<local_file_name>] [brief 1 full] [status=r11r21r31ef]

modification 1 mod 1 all : This Is a 1 to 9 character modification name. Alternatively a list of modification names may be given separated by commas and enclosed in parentheses or a range of modifications may be indicated by giving the first and last modification names separated by ellipses. Using the keyword ALL has the same effect as specifying the first and last modifications on the library as a range.

name 1 na : This is the name of a deck. If this parameter is specified, descriptive information will be given about only the named modification as it applies to this deck.

base I ba : This is the name of the file containing the base source library. If this parameter is not given an attempt is made to access a file with the name BASE.

list : I : This is the name of the file which receives the listing. If this parameter is not given, the file name OUTPUT will be used.

brief 1 br 1 full 1 fu : If the FULL keyword is included 43 on this command, the descriptive information about this 44 modification will be displayed as well as every line of 45 text altered by this modification. If the BRIEF keyword 46 is included or this parameter is omitted entirely, then 47 only the descriptive information about this modification 48

03/24/80

3.3.2 DISPLAY_MODIFICATION 1 DIM

×

3.0 COMMANDS

l=fixforb full	e		
In this example modification named along with any lines routineb.	psr123456	is listed on the	file FIXFORB
			·

Example: scu.display_modification mod=psr123456 na=routineb ba=syspl ...

is displayed.

status 1 st : See basic status concept.

ERS for Source Code Utility (NOS 170 Version) REV:J

3-41

03/24/80

COMPANY PRIVATE

1 5

1 2

8

5 6

1 6

1 7

1 8

1 9

110

111

113

14

115

116

17

18

19

120

121

122 23

124

125 26

127

128 29

130

131

132

133 34

135

base source library. If this parameter is not given an	136
attempt is made to access a file with the name BASE.	137
	38
result 1 r : This is the file to receive the modified	139
source library. If this parameter is not given, the	140
modified source library will be written to a file named	141
RESULT.	142
	43
state : This is the new state for the modification. For	145
a user to raise a modification to a given state or lower	145
it from that state to 0, he must have the corresponding	146
authority over the base source library or a higher one.	47
Values this parameter may assume are:	48

COMPANY PRIVATE

ERS for Source Code Utility (NOS 170 Version) REV: J 3.0 COMMANDS 3.3.3 CHANGE_MODIFICATION 1 CHM 3.3.3 CHANGE_MODIFICATION | CHM This command enters or modifies descriptive information about a modification. change_modification modification=<modification_name>tall [feature=<feature_name>] [author=<author_name_string>] -[description=<descriptive_string>] [base=<file_name>] [result=<file_name>]

[state=<integer>]

modifications on a library as a range.

to be associated with this modification.

[status=r11r21r3tef]

modification ; mod ; all ; This is a 1 to 9 character

modification name. Alternatively a list of modification

names may be given separated by commas and enclosed in

parentheses or a range of modifications to be changed may

be indicated by giving the first and last modification

names separated by ellipses. Using the keyword ALL has

the same effect as specifying the first and last

feature 1 fe : This is a 1 to 31 character feature name

author 1 au : This 1 to 31 character string can be used

description | de # This 1 to 256 character string can be

used to describe the modification. It is suggested that

this be used to point to official documentation for the

base 1 ba : This is the name of the file containing the

modification such as psr numbers or design documents.

to identify the individual who wrote this modification.

CDC SOFTWARE ENGINEERING SYSTEM

03/24/80

03/24/80

ce Code Uti	lity (NOS 170 Ver	'sion)	REV: J
	ON I CHM		
	UN 1 GANA Waxaalaa ahaalaa ahaalaa		
	om en tal		
3 : Verifi			
4 t Releas	ed		
tus I st = S	ee basic status o	:oncept.	
nymod de="sh	ort, fat, bald, b	eard" state=4	
• •			.*
		· · ·	
	0 : Experin 1 : Develor 2 : Stable 3 : Verific 4 : Release	0 : Experimental 1 : Developmental 2 : Stable 3 : Verified 4 : Released tus I st : See basic status o	<pre>MODIFICATION : CHM 0 : Experimental 1 : Developmental 2 : Stable 3 : Verified</pre>

D COMMANDS	
3.4 PURGE_NODIFICATION : PUN	~~~~~
3.3.4 PURGE_MODIFICATION 1 PUM	
This command is used to remove a modification from t	
source library. In order to purge a modification a user mu have the appropriate authority for the library to manipula	
modifications at this state. One will not be allowed to pur	
a modification that is associated with the creation of	
deck. Such decks must be purged first.	
purge_modification modification= <modification_name></modification_name>	
[name= <deck_name>]</deck_name>	. *
[base= <file_name>]</file_name>	
[result= <file_name>] [status=r1!r2!r3!ef]</file_name>	
modification 1 mod : This is the name of a modification	
Alternatively a list of modification names may be giv	
separated by commas and enclosed in parentheses or range of modifications may be indicated by giving t	
first and last modification names separated by ellipse	
name 1 na : This is the name of a deck. If th	ic
parameter is given, only the part of the modificati	
that applies to this deck will be removed. Alternative	•
a list of names separated by commas and enclosed	-
parentheses may be given or a range of decks may indicated by giving the first and last deck nam	
separated by ellipses.	
base 1 ba # This is the name of the file containing t	
base source library. If this parameter is not given attempt is made to access a file with the name BASE.	an
result 1 r : This is the file to receive the modifi	
source library. If this parameter is not given, t	
modified source library will be written to a file nam RESULT.	ea
status 1 st : See basic status concept.	
Example:	
scu.purge_modification fangs na=serpents ba=beasts r=animals	

3-45

12

3

6

1 5

: 6

1 7

16

17

18

19

20 21

22

23

24

25

28

27 28

29

130

131

132

134

135

136

137

139 48

141

14845678 45678

							03/24/80
						170 Version)	REV: J
				10 -10 40 10 10 10 40 40 40 4		na na an a	a an in an
		IANDS					
3.3.	•5 SE	EQUENCE	_MODI	FICATION	I SE	Maria da Cara d	

3.3.5 SEQUENCE_MODIFICATION 1 SEM

This allows a user to resequence a modification so that sequence numbers will appear in the order that source lines appear in the deck rather than in the order they were introduced. This is only allowed for modifications in state 0. All inactive lines introduced under this modification name will be discarded.

sequence_modification	modification= <modification_name></modification_name>
	[name= <deck_name>]</deck_name>
	[base= <file_name>]</file_name>
	[result= <file_name>]</file_name>
	[status=r11r21r31ef]

modification 1 mod * This is the name of a modification. Alternatively a list of modification names may be given separated by commas and enclosed in parentheses or a range of modifications may be indicated by giving the first and last modification names separated by ellipses.

name 1 na : This is the name of a deck. If this parameter is given, only the part of the modification which applies to this deck will be resequenced. Alternatively a list of names separated by commas and enclosed in parentheses may be given or a range of decks may be indicated by giving the first and last deck names separated by ellipses.

base 1 ba : This is the name of the file containing the base source library. If this parameter is not given an attempt is made to access a file with the name BASE.

result 1 r : This is the file to receive the modified source library. If this parameter is not given, the modified source library will be written to a file named RESULT.

status 1 st : See basic status concept.

Example: scu-sequence_modification psr12345 CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 3.0 COMMANDS 3.3.6 EXTRACT_MODIFICATION | EXM 3.3.6 EXTRACT_MODIFICATION 1 EXM 1 2 This command is used to extract a modification from a 1 5 source library in the form of the editor commands INSERT, 1 4 DELETE and REPLACE. 1 5 6 extract_modification modification=<modification_name> 7 8 edit_commands=<local_file_name> 1 9 [name=<deck_name>] [base=<file_name>] :10 [sculupdate] 111 [status=r11r21r31ef] 112 13 modification : mod : This is the name of a modification. 114 Alternatively a list of modification names may be given 115 separated by commas and enclosed in parentheses or a 116 range of modifications may be indicated by giving the 117 first and last modification names separated by ellipses. 118 19 edit commands 1 ec 1 This is the name of the local file 28 to receive the edit commands generated by this command. 21 22 name I na I This is a deck name. If specified only that 123 part of the modification which applies to this deck wilt 124 be written to the EDIT_COMMANDS file. 125 26 128 base I ba : This is the name of the file containing the base source library. If this parameter is not given an 128 attempt is made to access a file with the name BASE. 129 30 scu i update : If the UPDATE keyword is specified the 131 modification will be externalized as a six bit display 132 code UPDATE correction set containing *BEFORE, *INSERT 138 and *DELETE directives. If the SCU keyword is given or 134 this parameter is omitted entirely, the modification will 135 be externalized as the ASCII SCU editor commands INSERT. 136 DELETE and REPLACE. 137 38 status 1 st : See basic status concept. 139 40 Example: 141 scu.extract_modification fred ec=comfile ba=pets 142 43 44 45 48 47

48

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 3.0 COMMANDS 3.3.7 DISPLAY_MODIFICATION_LIST 1 DML 3.3.7 DISPLAY_MODIFICATION_LIST | DML 1 2 This command displays a list of all the modifications for a 3 source library. 6 5 display_modification_list [base=<file_name>] 1 6 [list=<local_file_name>] 1 7 [brief!full] 1 8 Istatus=r11r21r31ef] 1 9 10 base I ba : This is the name of the file containing the 111 base source library. If this parameter is not given an 112 attempt is made to access a file with the name BASE. 113 14 list 1 1 : This is the name of the file which receives 15 the listing. If this parameter is not given, the file 16 name OUTPUT will be used. 17 18 brief 1 br 1 full 1 fu 1 If the BRIEF keyword is included 19 on the command or this parameter is omitted a shorter 20 display will be given. If the FULL keyword is present a 21 more detailed display will be given. Currently the 22 displays selected by the two keywords are identical. 23 24 status 1 st : See basic status concept. 25 26 127 Example: SCU.DML library modilst 128 29 30 31 3.3.8 DISPLAY_FEATURE 1 DIF 32 This command is used to produce a modification description 33 for each modification associated with a feature. 34 35 display_feature_feature=<feature_name> 136 [base=<file_name>] 137 [list=<local_file_name>] 138 [brief 1 full] :39 [status=r11r21r31ef] 148 41 feature : fe : This is the name of the feature to be 42 displayed. 43 45 base ; ba : This is the name of the file containing the 145 base source library. If this parameter is not given an 146 attempt is made to access a file with the name BASE. 147

COMPANY PRIVATE

48

3-48 CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REVIJ 3.0 COMMANDS 3.3.8 DISPLAY FEATURE ! DIF list 1 | : This is the name of the file which receives 2 the listing. If this parameter is not given, the file 2 name OUTPUT will be used. 8 G brief 1 br 1 full 1 fu 1 If the BRIEF keyword is included 5 on this command or this parameter is omitted entirely. 6 the display will include only a list of the modifications . associated with the feature. If the FULL keyword is 8 9 used, the display will give the modification descriptions for each of these modifications as well. 10 11 status 1 st : See basic status concept. 12 18 Example: 114 scu.display_feature enhanced_network_interface 115 16 17 3.3.9 DISPLAY FEATURE LIST I DFL 18 19 This command produces a list of all feature names for a 28 source library and optionally a list of which modifications 21 are associated with each. 22 23 display_feature_list [base=<file_name>] 124 [list=<local_file_name>] 125 [trief [full] 126 [status=r11r21r31ef] 127 28 base 1 ba: This is the name of the file containing the 129 base source library. If this parameter is not given an :3# attempt is made to access a file with the name BASE. 131 32 list 1 1 : This is the name of the file which receives 33 the listing. If this parameter is not given, the file 34 name OUTPUT will be used. 35 36 brief 1 br 1 full 1 fu : If the BRIEF keyword is included 37 on this command or this parameter is omitted entirely, 38 only the list of feature names for this source library 39 will be displayed. If the FULL keyword is used, lists of 40 the modifications associated with each feature will be 4**1** included as well. 42 43 status 1 st : See basic status concept. 144 45 146 Example: scu.display_feature_list myop1 147 48

12

8

1 6

: 5

: 6

17

1 8

1 9

118

111

:12

113

14

15

116

117

18

119

120

121

122

123

124

125

126

127

128

129

130

132

133

134

136

:38

136

139 148

41

142

143

03/24/80

ERS for Source Code Utility (NOS 170 Version) REV:J

3.0 COMMANDS

3.4 CONVERSION AIDS

3.4 CONVERSION AIDS

Two commands are available to convert source code from UPDATE and MODIFY program library format to SCU source library format. Update and Modify names which do constitute valid SCU names can be preserved as will sequence information. All modifications are set to 0 (experimental) state. This is done to allow a user the opportunity to add descriptive information to key modifications. On conversion *WEOR directives will be changed to *WEOP directives. Because only one level of file separator will be supported under NOS/VE, the *WEOF directive will be changed to a *WEOP directive as well and the user will be informed.

A command is available to convert source code from SCU source library format to UPDATE program library format. With the exception of name substitution, sequence information will be preserved.

Deck names or correction identifiers which do not qualify as valid SCU deck or modification names will be flagged as errors and the user will be expected to effect name substitution during a subsequent attempt at conversion. Put another way, these conversion utilities will reject names that do not begin with a letter or \$.

Buring conversion a user may wish to make name substitutions either because existing names on the UPDATE or MODIFY libraries may not qualify as SCL names or because he wishes to use the longer names allowed by SCU. The conversion utilities allow for name substitution through the NAME_LIST file. Each line in this file is interpreted as the parameters of a SCL command as shown below:

```
nos_name=<update_or_modify_name>
[scl_short=<modification_name>]
[scl_long=<deck_name>l<define_name>]
```

nos_name : This is a the name of an UPDATE or MODIFY deck, correction identifier or DEFINEd name. <u>The user</u> <u>should specify this parameter with the keyword if the</u> <u>name contains characters other than letters</u>, <u>digits and</u> <u>\$.</u>

scl_short : This is a 1 to 9 character SCU modification name.

scl_long:This is a 1 to 31 character SCU deck name or145DECLARE name (see criteria file directives).If this146parameter is omitted the value given for SCL_SHORT will147be used.If a value is not given for at least one of the148

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 3.0 COMMANDS 3.4 CONVERSION AIDS parameters SCL_SHORT or SCL_LONG an error will be 1 1 reported and conversion inhibited. 1 2 3 Example: 1 4 dsd dsd dynamic_display_driver 1 5 zscmere scm\$ere scm\$replace 1 6 nos_name=old=fool old_fool 1 7 1 8 On the first line the MODIFY name DSD will continue to be 1 9 used on SCU as the modification name associated with lines 118 introduced under that name, but the deck name will have the 111 longer, more descriptive form. On the second line a MODIFY 112 deck name is being changed to fit the NOS/VE naming pattern. 118 On the third line the name OLD=FOOL is being changed to 114 OLD_FOOL to make it a legal SCL name. 115 16 17 16 3.4.1 CONVERT_UPDATE_TO_SCU I CUTS 19 28 This command converts an UPDATE program library to an SCU 21 22 source library. 28 convert_update_to_scu {oldp1=<local_file_name>} 124 [result=<file_name>] 125 [list=<local_file_name>] 26 [name_list=<local_file_name>] 27 [brief | full] 28 [ascii64 | ascii812] 29 [criteria=<local_file_name>] 130 [status=r11r21r31ef] :31 32 oldpl : This is the name of a file containing an UFDATE 33 program library in sequential format. If this parameter 134 is not given an attempt will be made to access a local 135 file with the name OLDPL. 136 37 result : r : This is the file to receive the new source 138 library. If this parameter is not given, the new source :39 library will be written to a file named RESULT. 140 41 42 list 1 1 This is a file to receive a report describing 143 the conversion. The default value for this parameter is 144 OUTPUT. 145 46 name_list ! nl : See discussion of name substitution :47 above. 148

NO COSTUARE ENCINEERING EVETEN	3-51
DC SOFTWARE ENGINEERING SYSTEM	03/24/80
RS for Source Code Utility (NOS 170 Version) REV:	J ~~~~~
.0 COMMANDS .4.1 CONVERT_UPDATE_TO_SCU I CUTS	***
brief 1 br 1 full 1 fu : If the BRIEF keyword is in	cluded
on the command or this parameter is omitted, a s	
form of the listing will be produced. If th keyword is used a more detailed listing wil produced.	
ascii64 ascii812 : These alternate keywords info	rm the
conversion routine whether the input program libra	
in the 64 character set or full ascil. ASCII selected if neither keyword is given. <u>Currentl</u>	
conversion from ASCII update (ASCII812) i implemented.	
criteria cr = This is the name of a file to r	
criteria file directives which will preserve a p	
the information from the YANK\$\$\$ deck on the program library. Active YANK*s will be converted	
corresponding EXCLUDE's. Active DEFINE's wit	
converted to DECLARE's. Inactive cards in the Y	
deck will be ignored. If this parameter is omitte contents of the YANK\$\$\$ deck will be ignored.	d the
status i st i See basic status concept.	
Example:	
scu.convert_update_to_scu pl1a scupl1a saga ascii64	
3.4.2 CONVERT_MODIFY_TO_SCU 1 CMTS	
This command converts a MODIFY program library to	an SCU
source tibrary.	
convert_modify_to_scu [op1= <local_file_name>]</local_file_name>	
[result= <file_name>]</file_name>	
[list= <local_file_name>]</local_file_name>	
[name_list= <local_file_name>]</local_file_name>	
[brief f full] [ascii64 1 ascii612]	
[ascilo4 ascilo12] [criteria= <tocal_fite_name>]</tocal_fite_name>	
[status=r11r2ir3ief]	
op! # This is the name of a file containing a	MODIFY
program library. If this parameter is not gi	ven an
attempt is made to access a local file named OPL.	
result 1 r : This is the file to receive the new	source
COMPANY	PRIVATE

COMPANY PRIVATE

~	-	~	1.00	~	n

	03/24/80
ERS fo	r Source Code Utility (NOS 170 Version) REV:J
	MMANDS
5.4.2	CONVERT_MODIFY_TO_SCU 1 CMTS
	library. If this parameter is not given, the new source library will be written to a file named RESULT.
	list 1 1 : This is a file to receive a report describing
	the conversion. The default value for this parameter is OUTPUT.
	001 - 01 •
	name_list : nl : See discussion of name substitution above.
	brief t br t full t fu t If the BRIEF keyword is included
	on this command or this parameter is omitted, a shorter
	listing will be produced. If the keyword FULL is
	included a more detailed listing will be produced.
	ascii64 1 ascii612 # These alternate keywords inform the
	conversion routine whether the input program library is
	in the 64 character set or full ascii. ASCII612 is
	selected if this parameter is omitted.
	criteria 1 cr : This is the name of a file to receive
	criteria file directives. As an example, modifications
	flagged as yanked by MODIFY will be entered into the
	criteria file as EXCLUDE_MOD directives.
	status I st I See basic status concept.
Eva	mple:
	.convert_modify_to_scu rel1a scu1a saga ascli64
-	
5.4	.3 CONVERT_SCU_TO_UPDATE 1 CSTU
	This command converts an SCU source library to an UFDATE
	gram library. <u>This command is not currently implemented</u>
and	the requirement for such a command is under review.
con	vert_scu_to_update {base= <file_name>}</file_name>
11	[newpl= <local_file_name>]</local_file_name>
	[list= <local_file_name>]</local_file_name>
	[name_list= <local_file_name>]</local_file_name>
	[brief full] [ascii812 ascii64]
	<pre>(asciloi2 / ascilo4) (criteria=<10cal_file_name>)</pre>
	[status=r11r21r31ef]
	base 1 ba # This is the name of a file containing an SCU

03/24/80

s for	Source Code Utility (NOS 170 Version)	REV: J
	IANDS INVERT_SCU_TO_UPDATE I CSTU	
	source library to be converted. If this par given an attempt will be made to access a BASE.	
	newpl : This is the file to receive to program library. If this parameter is not o program library will be written to a file no	given, the new
	list 11: This is a file to receive a report the conversion. The default value for this OUTPUT.	
	name_list nl : See discussion of name above.	e substitution
	brief 1 br 1 full 1 fu : If the BRIEF keywor on the command or this pararmeter is omit listing will be produced. If the FULL included a more detailed listing will be give	ted, a shorter . keyword Is
	asci1812 asci164 : These alternate keywor conversion routine whether the output progra in the 64 character set or full ascil. ASC if this parameter is omitted.	am library is
	criteria 1 cr : This is the name of a cr DECLARE directives will appear as *DEFINE the YANK\$\$\$ deck on NEWPL. EXCLUDE dir appear as *YANK *SELYANK and *YANKDECK direc	directives in Pectives will
	status I st : See basic status concept.	
Exam scu.	ole: convert_scu_to_update scuplia plia myth	

ERS for	03/24/ Source Code Utility (NOS 170 Version) REV:J
3.0 COMI 3.5 CALI	1ANDS ING THE EDITOR
3.5	CALLING THE EDITOR
edito unti	ne command described in this section is used to call the or. All commands will be interpreted as editor commands one exits the editor. The editor can be used in either or interactive mode.
3.5.	LEDIT
	his command establishes an environment for the other line or commands.
edit	<pre>modification=<modification_name> [base=<file_name>] [result=<file_name>] [input=<local_file_name>] [list=<local_file_name>] [name=<deck_name>] [continue] [status=r11r2tr31ef]</deck_name></local_file_name></local_file_name></file_name></file_name></modification_name></pre>
	modification 1 mod : This is the name for the modification being introduced or altered.
	base I ba : This is the name of the file containing a source library containing one or more decks to be edited. If this parameter is not given an attempt is made to access a file with the name BASE.
	result i r : This is the name of the file to receive the modified source library. If this is not given the modified library will be written to a file named RESULT.
	input 1 i : This is the name of the file from which edit commands are to be read. The default value is INPUT.
	list 1 1 : This is the name of the file which receives the listing. If this parameter is not given, the file name OUTPUT will be used.
	name 1 na * This is the name of the deck to edit first.
	continue : con : If this keyword is present, the editor will allow the user to add to an existing modification in state D.

COMPANY PRIVATE

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 3.0 COMMANDS 3.5.1 EDIT status 1 st : See basic status concept. Example: scu.edit improve oldp1 continue

: 1

1 3

12

3

5

5

1 6

1 7

13

15

15

16

18

18

19 28

21

122

124

125

126

28 129

:38

131

132

135

135

136

138

139

140

141 42

143

144

145 46 47

48

03/24/80

ERS for Source Code Utility (NOS 170 Version) REV:J

3.D COMMANDS

3.6 GROUPING SCU COMMANDS

3.6 GROUPING SCU COMMANDS

Although SCU commands are available as separate job control statements that may be used entirely independently of one another, it is often desirable to associate them with one another to make a series of changes to a base source library and produce a single result source library and listable output. This is done by using the GROUP and END commands.

3.6.1 GROUP

This command is used to flag that following SCU commands are to be associated with one another and to assign default values for the names of some of the files that these commands must access. When this command is used the base source library is copied to a working copy. Subsequent SCU commands using the default base source library will modify this working copy. Using the END command causes the working copy to be written to the result source library. Nondefault base and result source libraries may be named explicitly on individuat commands.

Permanent files named on the BASE and RESULT parameters on the GROUP command will remain local until the corresponding END command is executed. The commands listed above under conversion aids should not be used in group mode.

group (base=<file_name>]
 [result=<file_name>]
 [list=<local_file_name>]
 [status=r11r21r31ef]

base I ba : This is the name of the file containing the base source library. If this parameter is not given an attempt is made to access a file with the name BASE.

result 1 r : This is the file to receive the modified source library. If this parameter is not given, the modified source library will be written to a file named RESULT.

list 1 1 : This is the name of the file which receives the listing. If this parameter is not given, the file name OUTPUT will be used.

Example: See example below under end.

COMPANY PRIVATE

3-57 03/24/80

ERS	for	Source	Code	Utility	(NOS	170	Version)	REV: J	
7 0	000	44400							

3.0 COMMANDS 3.6.1 GROUP

1 2 3 3.6.2 END 4 5 This command causes the working copy of the source library 6 7 to be written to the result source library file and any subsequent SCU commands to operate independently of one 8 9 another. 18 end [status=<status_variable>] 11 12 13 Example: scu.group oldlib newlib 1st0212 115 scu.display_library full 115 attach.textfp. 16 scu.create_deck ((fetch_parameters,textfp)) .. 117 fetch_p g=scu 118 scu.create_modification mod_to_4 au= *M. J. Perreten* . 119 20 de="Adds syntax check." scu.edit mod_to_4 continue i=edcoms 121 scu.display_library_alterna_l=altlist 122 scu.change_modification axle state=4 123 scu.display_library full 125 scu.end 125 26 In this example the GROUP command declares that the SCU 127 commands that follow will be associated with one another. 128 OLDLIB is the file which contains the base source library. 29 All changes to this base are accumulated on a working copy and 38 will be written to the result source library NEWLIB when an 31 END command is encountered. Displays will be written to the 32 file LST0212 unless another list file is explicitly named on 33 34 an individual command. DISPLAY_LIBRARY command gives a detailed 35 The first description of the original source library. 36 The attach of file TEXTFP is shown to illustrate that other 37 commands can be intermixed with SCU commands. 38 The CREATE_DECK command creates a new deck on the working 39 copy of the source library using the contents of file TEXTFP 49 as the initial text. 41 The CREATE_MODIFCATION command creates a new modification 42 on the working source library. 43 The EDIT command calls the editor to perform the edit. 144 commands from file EDCOMS on the working copy of the source :45 library. Lines introduced by this session will be part of the 146 modification mod_to_4. 47 The second DISPLAY_LIBRARY command is included to emphasize 48

03/24/80

3-58

for	Sou	rce	Code	eU	†11	i ty	(NO	S 17	0 Ver	SIO	n)	~~~~	~~~~	RE	٧: .)	****	~~~~
		s ~~~~		-	***	*** ***						-	~~~~	-	-			
to ma Th nodii (rele (rele Th descr chang Th Th	anip he fica ease he fipt ges he ten	ulat CHA tion d). las ion intr END to t	e. NGE ax t of oduc con he i	_MO le DI th ced mma	DIF on SPL by nd ult	n ds C AT t he A Y Wor th ca so	can ION Wor LIBR king e en uses urce	exp co king ARY sou tire th tib	licit mmand sour com urce bloc e wor rary	ly ce mane lii k o kin on	name rais lib d brar f SC g so file	a es rar giv y U c urc NE	the y t es afte omma e 11 WLIB	a nds bra	de al	te ate tai to fl	of 4 led the be ags	
anott	ner.																	
,																		
	COMI 2 EI 2 EI 10 ma 10 ma 11 10 ma 11 10 10 ma 11 10 10 ma 11 10 10 10 10 10 10 10 10 10 10 10 10	COMMAND 2 END 2 END 2 END 2 END 2 END 3 EN	COMMANDS 2 END hat individ to manipulat The CHA nodification (released). The las description changes intr The END written to t that subsequ	COMMANDS 2 END hat individual o manipulate. The CHANGE nodification ax (released). The last description of changes introduc The END convitten to the that subsequent	COMMANDS 2 END hat individual co to manipulate. The CHANGE_MO nodification axle (released). The last DI description of th changes introduced The END comma written to the res that subsequent SC	COMMANDS 2 END hat individual comma o manipulate. The CHANGE_MODIF modification axle on (released). The last DISPL description of the changes introduced by The END command written to the result that subsequent SCU c	COMMANDS 2 END hat individual commands o manipulate. The CHANGE_MODIFCAT modification axle on the (released). The last DISPLAY_ description of the wor changes introduced by th The END command ca written to the result so that subsequent SCU comm	COMMANDS 2 END that individual commands can to manipulate. The CHANGE_MODIFCATION modification axle on the wor (released). The last DISPLAY_LIBR description of the working changes introduced by the en The END command causes written to the result source that subsequent SCU commands	COMMANDS 2 END that individual commands can explo- manipulate. The CHANGE_MODIFCATION com modification axle on the working (released). The last DISPLAY_LIBRARY description of the working som changes introduced by the entire The END command causes the written to the result source libe that subsequent SCU commands will	COMMANDS 2 END that individual commands can explicit to manipulate. The CHANGE_MODIFCATION command modification axle on the working sour (released). The last DISPLAY_LIBRARY com description of the working source changes introduced by the entire bloc The END command causes the wor written to the result source library that subsequent SCU commands will exe	COMMANDS 2 END that individual commands can explicitly to manipulate. The CHANGE_MODIFCATION command modification axle on the working source (released). The last DISPLAY_LIBRARY command description of the working source lin changes introduced by the entire block of The END command causes the working written to the result source library on that subsequent SCU commands will execute	A 2 END that individual commands can explicitly name to manipulate. The CHANGE_MODIFCATION command rais modification axle on the working source lib (released). The last DISPLAY_LIBRARY command description of the working source librar changes introduced by the entire block of SC The END command causes the working so written to the result source library on file that subsequent SCU commands will execute in	COMMANDS 2 END that individual commands can explicitly name a manipulate. The CHANGE_MODIFCATION command raises modification axle on the working source library (released). The last DISPLAY_LIBRARY command give description of the working source library changes introduced by the entire block of SCU co The END command causes the working source written to the result source library on file NE	COMMANDS 2 END that individual commands can explicitly name alter to manipulate. The CHANGE_MODIFCATION command raises the modification axle on the working source library t (released). The last DISPLAY_LIBRARY command gives description of the working source library afte changes introduced by the entire block of SCU comma The END command causes the working source li written to the result source library on file NEWLIB that subsequent SCU commands will execute independe	COMMANDS 2 END that individual commands can explicitly name alternat o manipulate. The CHANGE_MODIFCATION command raises the s modification axle on the working source library to (released). The last DISPLAY_LIBRARY command gives a description of the working source library after changes introduced by the entire block of SCU commands The END command causes the working source library written to the result source library on file NEWLIB and that subsequent SCU commands will execute independent	COMMANDS 2 END that individual commands can explicitly name alternate to manipulate. The CHANGE_MODIFCATION command raises the star modification axle on the working source library to sta (released). The last DISPLAY_LIBRARY command gives a de- description of the working source library after all changes introduced by the entire block of SCU commands. The END command causes the working source library written to the result source library on file NEWLIB and that subsequent SCU commands will execute independently of	COMMANDS 2 END that individual commands can explicitly name alternate fi to manipulate. The CHANGE_MODIFCATION command raises the state modification axle on the working source library to state (released). The last DISPLAY_LIBRARY command gives a detai description of the working source library after all changes introduced by the entire block of SCU commands. The END command causes the working source library to written to the result source library on file NEWLIB and fil that subsequent SCU commands will execute independently of	COMMANDS 2 END that individual commands can explicitly name alternate files to manipulate. The CHANGE_MODIFCATION command raises the state of modification axle on the working source library to state 4 (released). The last DISPLAY_LIBRARY command gives a detailed description of the working source library after all the changes introduced by the entire block of SCU commands. The END command causes the working source library to be written to the result source library on file NEWLIB and flags that subsequent SCU commands will execute independently of one

ERS for Source Code Utility (NOS 170 Version) REV:J 4.0 DIRECTIVES

4.0 DIRECTIVES

Directives are available to the SCU user to assist him in defining the contents of an expanded text file. These are text embedded directives intermixed with the source data and criteria file directives. This section describes both types of directives and their interaction with one another.

4.1 TEXT EMBEDDED DIRECTIVES

A text embedded directive (TED) will be recognized by the occurrence of a key character as the first character of a source line followed immediately by a keyword. If a TED is encountered the normal expansion will be altered in the mannner specified by the particular TED.

While processing the EXPAND_DECK command, TED's are processed and they are not copied into the expanded text file. While processing the EXTRACT_DECK command, TED's are not processed and are written to the the extracted text file.

4.1.1 COPY

This directive must be followed by a deck name. This directive will copy the text of the specified deck into the expanded text file. The same search order will be used as for decks listed on the EXPAND_DECK command. A deck may copy a deck that copies another deck and so on as long as the sequence is not recursive.

*copy name=<deck_name>

name 1 na : This is the name of a deck. Alternatively a list of names may be given separated by commas and enclosed in parentheses.

Example: *copy deck1

123456

7

8

9

10

11 12 14

14 15

16

17

18

19

20 21

22

23

24

29 38

31

32

33

34

35

36 37

138

:40

141

142

DC SUFTWARE ENGINEERING SYSTEM	03/24/80
RS for Source Code Utility (NOS 170 Version)	
.0 DIRECTIVES .1.2 COPYC (COPY CONDITIONAL)	
4.1.2 COPYC (COPY CONDITIONAL)	
This directive serves the same function as COP the text of the deck is written to the expande conditionally. In other words, if the specified copied into the expanded text file for the de- being expanded, it will not be included again.	d text file deck has been
*copyc name= <deck_name></deck_name>	
name I na # This is the name of a deck. Al list of names may be given separated by enclosed in parentheses.	
Example: *copyc deck1	
4.1.3 DECK	
This text embedded directive will only be it	nterpreted as
such when it occurs as the first line of a parti	tion (record
on NOS 170) in the file named by the NAMES_ON_F of the create deck command. Its purpose is to	
deck name for the following text lines.	provide the
*deck_name= <deck_name> [expandino_expand]</deck_name>	
name 1 na : This is the name to be give	n to the deck
which will contain the text lines that directive.	follow the
expand 1 exp 1 no_expand 1 nexp 1 If	the NO EVOING
keyword is included on the directive this de	
be written to the expanded text file pr	oduced by the
EXPAND_DECK command as a result of processi	
embedded directives *COPY and *COPYC. keyword is included or this parameter	
entirely, this deck will be written to the	
file by the EXPAND_DECK command whether n	amed on the
command or on the *COPY and *COPYC directive	5.
4.1.4 ELSE	
This setting disadius and be used to see	
This optional directive may be used to separate source data that are to be expanded if the given of	
true from those expanded if it is false. Either	
between the IF and the ELSE or between the ELSE a	

123

4 5

6

78

1 8

1 9

110

11

12

13

14

15 16

17

18 19

20

21

22 23

24 25

26 27 28

29

30 31

132

138

134

136

137

39

140

141

142

46

45

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J

4.0 DIRECTIVES 4.1.4 ELSE

may be empty.

4.1.5 IF

This directive is used to allow source lines to be written to the expanded text file conditionally. If the specified condition is true, the source lines between this directive and the next ELSE or IFEND directive (whichever occurs first) will be written to the expanded text file. Text embedded directives in the same range will be processed while processing the EXPAND_DECK command. If an ELSE directive is present before the IFEND the source lines between those two directives will be skipped as well as any text embedded directives.

If the specified condition is false, source lines and text embedded directives will be skipped until either an ELSE or IFEND directive is encountered. Source lines and text embedded directives between a following ELSE and IFEND will be processed. Normal (unconditional) processing resumes following the IFEND whether the condition is true or false.

There are two forms of this directive shown below.

*if \$scu_type(name,type)

name : This is a name to be checked for the attribute described in the type field.

type : This field is used to specify one of three keywords. Any of these keywords may be prefixed by NOT. The keywords are:

- DECK This is a name of a deck on any of the libraries specified by the BASE parameter.
- MOD This is the name of a modification to any of the libraries specified by the BASE parameter.
- DECL This is a name defined by a DECLARE directive on the criteria file associated with this EXPAND_DECK command.

ERS	for	Source C	ode Utili	ty (NOS 1	Version)	03/24/80 REV=J
	DIRI 5 II	ECTIVES	, nu au au au au au au au ao ne ne			
		Example:	These so is a dec *else	k name.	deck) 5 are expande 5 are expande	
			lines ar		e and these s if dsd is a	
				ional prod urce line:	cessing resum	ies with
		Francia	21 fif te	cu type (f)	Lx2,notmod)	
		T VOIN NI C			expanded if	there is no
			modifi	cation to	the source l	
				me fix2.		
			*ifend			
		*if \$ro(name,valu	e)		
		on EXP	the c AND_DECK	riteria	file assoc Testing a r	DECLARE directive lated with this dame that is not
			t This ues aret	is a r	elational ope	rator. acceptable
			gt 🖁 G	reater tha	an.	
			ge :G	reater tha	an or equal t	0.
				ess than.		
				ess than o	or equal.	•
				qual to. ot Equal 1	to	
			ue : This 65535.	is a pos	itive decimal	value between OF
		Exampte:		nes will I	be expanded l Je other than	
	C	onditiona		may be n	nested to a m	aximum depth of 32

	4-5
CDC SOFTWARE ENGINEERING SYSTEM	03/24/80
RS for Source Code Utility (NOS 170 Version)	REV: J
••0 DIRECTIVES ••1•5 IF	
levels with the restriction that every IF must have IFEND within the same deck.	a matching
4.1.6 IFEND	
This directive marks the end of a block of sour be expanded conditionally. An IFEND must occur in deck as the corresponding IF.	
4.1.7 TEXT	
This directive may be used to indicate that the block of source data is to be considered as text. words no scan will be made for text embedded direct for TEXTEND.	In other
4.1.8 TEXTEND	
This directive tells SCU to resume scaning source text embedded directives.	e data for
4.1.9 WEOP	
This directive will cause an end of partition to on the file receiving the expanded text. (In the Implementation of SCU, an end of record will be writ	e NOS 170
4.1.10 WEOPC	
This directive will cause an end of partition to to the file receiving the expanded text, if any so has been written to the file since the last separ the Cyber 170 implementation, an end of record written.)	ource data rator. (In

UC SUFIWARE ENGINEERING SYSTEM
03/24/8 RS for Source Code Utility (NOS 170 Version) REV:J
.0 DIRECTIVES .2 CRITERIA FILE DIRECTIVES
4.2 CRITERIA FILE DIRECTIVES
These directives are available to allow the SCU user to be more selective about what source data is written to the expanded text file. They are given on a separate file and are not preceded by a key character. They are processed in order of occurence.
4.2.1 DECLARE
This directive is used to define a name with a value associated with it that can be used in connection with the IF text embedded directive to conditonally include source data on the expanded text file.
declare name = <name> [value=<integer>]</integer></name>
name : na : This is a 1 to 31 character name.
value : This optional parameter is an integer in the range 0 to 65535 decimal. If this is omitted the value 0 is assigned.
Example: declare fritz 35
4.2.2 EXCLUDE_DECK
This directive can be used to explicitly exclude a particular deck from being written to the expanded text file.
exclude_deck_name= <deck_name></deck_name>
name I na I This is the name of a deck to be excluded. Alternatively a list of names separated by commas and enclosed by parentheses may be given or a range of decks may be indicated by giving the first and last deck names to exclude separated by ellipses.
4.2.3 EXCLUDE_FEATURE
This directive is used to exclude incomplete or unstable features from an expanded text file.

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 4.0 DIRECTIVES 4.2.3 EXCLUDE FEATURE 1 feature 1 fe : This is the name of a feature to exclude 1 2 when generating an expanded text file. Alternatively a : 3 list of features may be given separated by commas and 4 enclosed in parentheses. 5 6 * state : This parameter may be given as one of five integers 0-4 8 (see states in the concepts section). All modifications 9 which are part of this feature which are at or below the 10 given state will be excluded from the expanded text 11 file. If this parameter is omitted all modifications 12 associated with this feature will be excluded. 13 14 Example: exclude_feature new_network_interface 0 15 16 17 4.2.4 EXCLUDE_GROUP 18 19 This directive is used to exclude an entire group of 28 related decks from the expanded text file. 21 22 exclude_group group=<group_name> 123 24 group 1 g # This is a 1 to 31 character group name. 125 Alternatively a list of group names may be 126 given separated by commas and enclosed in parentheses or a 127 range of groups may be indicated by giving the first and 128 last group names to exclude separated by ellipses. 129 30 31 4.2.5 EXCLUDE_MOD 32 33 This directive can be used to exclude the effects of a 34 particular modification. 35 36 modification=<modification_name> 137 exclude_mod [name=deck name] 138 39 modification 1 mod : This is the name of a modification 148 to be excluded from the expanded text 141 file. Alternatively a list of modifications may be 42 given separated by commas and enclosed in parentheses. 43 44 name 1 na : This is the name of a deck. That part of the 145 modification which applies to this deck will be excluded. 146 on the expanded text file. Alternatively a list of deck 47 names may be given separated by commas and enclosed in 48

C COETWADE ENCINEEDING SYSTEM	4-8
C SOFTWARE ENGINEERING SYSTEM	03/24/80
S for Source Code Utility (NOS 170 Version)	REV:J
0 DIRECTIVES 2.5 EXCLUDE_MOD	
parentheses.	
Example: exclude_mod newmethod obsolete_deck	
4.2.6 EXCLUDE_STATE	
This directive is used to exclude the modifications at or below a given state.	effects of all
exclude_state state= <integer></integer>	
state : This parameter is given as one of 0-3.	the integers
4.2.7 INCLUDE_COPYING_DECKS	
This directive causes all decks which copy a be written to the expanded text file. Chains references will be followed until decks wi attribute are found.	s of indirect
include_copying_decks_name= <deck_name></deck_name>	
name I na # This is a deck name. As on th command the name parameter may be give range of decks.	
4-2-8 INCLUDE_DECK	
This directive is used to specifcally include deck on the expanded text file.	e a particular
include_deck name= <deck_name></deck_name>	
name 1 na : Th is is a deck name. As on t command the name parameter may be given a range of decks.	
4.2.9 INCLUDE_FEATURE	
This directive is used to include all the associated with a feature which are at a given son the expanded text file.	

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 4.0 DIRECTIVES 4.2.9 INCLUDE_FEATURE include feature=<feature_name> [state=<integer>] 1 1 2 feature 1 fe : This is the name of a feature to be 1 3 included. Alternatively a list of features may be given 1 4 separated by commas and enclosed in parentheses. 5 6 state : This optional parameter is given as one of five 7 integers 0-4 (see states in the concept section). If 1 8 this parameter is omitted all modifications associated 1 9 with this feature will be included. 110 11 Example: include_feature new_network_interface 3 12 13 14 4.2.10 INCLUDE_GROUP 15 15 This directive is used to cause an entire group of related 17 decks to be included on the expanded text file. 18 19 120 include_group group=<group_name> 21 group 1 g : This is a 1 to 31 character group name. 122 Alternatively a list of group names may be 123 aiven separated by commas and enclosed in parentheses or a 124 range of groups may be indicated by giving the first and 125 126 last group names to include separated by ellipses. 27 28 4.2.11 INCLUDE_MOD 29 38 This directive is used to explicitly include a modification 31 in the expanded text file. 32 33 include mod modification=<modification_name> 134 [name=<deck_name>] 135 36 modification 1 mod # This is the name of a modification 137 to be included. Alternatively a list of modifications my 138 be given separated by commas and enclosed in 39 parentheses. 40 41 name I na : This is the name of a deck. 142 If this parameter is given only that part of the modification 143 that applies to this deck will be included. As on the 46 EXPAND_DECK command the name parameter may be given as a 45 list or range of decks. 46 47 48 Example: include_mod accounting_fixes

COMPANY PRIVATE

03/24/80

			•			Version)	REV: J
4.0	DIRE	ECTIVES		, 40 40 40 40 4	~ ~ ~ ~ ~		đượ đảo
		INCLUDE	 10 M2	~~~~~		بله جاد ماه چه الله مه چه چه وا	میں خوا میں جو میں ہیں ہیں ہیں جو میں ہیں جو میں ہی جو میں اور میں میں میں میں ہیں ہیں ہیں ہیں ہیں ہیں

4.2.12 INCLUDE_STATE

This directive is used to include the effects of all modifications at or above a given state.

include_state state=<integer>

state : This parameter is given as one of the integers D-4.

123456

8

8

9

10 11 12

13 14

15

16

17 18 19

2021

128

23

24

25

126

127

128

29 31

31

32 33 34

35 36

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 5.0 SCU EDITOR

5.0 SCU EDITOR

The SCU editor provides the capability of generating and manipulating source text. It operates on files in source library format allowing source lines to be introduced or corrected either interactively or in batch mode.

5.1 EDITOR CONCEPTS

This section introduces some concepts that are used in the SCU editor. Although most of the terms are not new they may have particular meaning in the SCU context.

5.1.1 TABBING

The TAB command defines a tab character and associated tab columns. Other commands which involve inserting text scan the inserted text for the tab character. When the tab character is encountered, spaces are inserted into the text from that point up to the next tab column where the next character is placed. The tab character is not placed in the library text. One can define a default value for the tab character and columns in each deck header.

One tab character and up 256 associated tab columns can be active at one time.

5.1.2 LINE IDENTIFIERS

Each line within an SCU library is assigned a line 137 identifier which is unique within a deck. This identifier 138 consists of the 1 to 9 character modification name and a six 39 digit sequence number. The sequence number will start from 1 48 as the first line introduced under the modification in a deck 4 £ and increment by 1 as lines are added. Continuing a 42 modification will result in sequence numbers starting 1 43 greater than the last previously introduced to this deck. 44 Sequencing starts from 1 in each deck rather than being 45 continuous throughout the source library. To reference a line 46 by line identifier one gives the modification name separated 47 from the sequence number by a period. Trailing blanks in the 48 CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 5.0 SCU EDITOR 5.1.2 LINE IDENTIFIERS modification name and leading zeros in the sequence numbers 膏 will have no significance. 2 3 6 5.1.3 KEYWORD VALUES 5 6 On some editor commands, lines or position within a deck 1 7 may be given through the use of keywords. These keywords and 1 8 their short forms are listed below. 9 19 ALL 1 A : This indicates all active lines in an SCU 111 deck. 112 13 FIRST : F : This indicates the first active line in an 114 SCU deck. 115 16 LAST ! L : This indicates the last active line in an SCU 117 deck. 118 19 120 CURRENT 1 C : This indicates the current active line in an SCU deck. 121 22 On those line editor commands using the number parameter the 28 keyword below can also be used to specify a value. 24 25 ALL ! A : This specifies every active line in a range. 126 27 On those line editor commands using the occurrence parameter, 128 the keyword below can be used to specify a value. 129 30 LAST | L : This specifies the last occurrence of the line 131 or block. 132 33 On the DECK command a deck to be edited can be selected by 134 giving a deck name explicitly or by the use of the keywords 135 below. 136 37 FIRST : This keyword selects the first deck in the 138 directory on the library as the object of the commands 139 which follow. :40 41 LAST : This keyword selects the last deck in the 142 directory on the library as the object of the commands 143 which follow. 144 45 NEXT : This command selects the next deck in the 146 directory on the library as the object of the commands 147 which follow. 148

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 5.0 SCU EDITOR 5.1.4 WIDTH

5.1.4 WIDTH

Width is a maximum number of characters to be allowed in a text line. Line overflow occurs if the user causes a line to be longer than the width declared. The editor issues a warning message and allows the overflowing line to be added or altered. It is the user's responsibility to split longer lines where necessary.

5.1.5 THE BATCH USER

The SCU line editor will in some cases discriminate between batch and interactive users. The VETO option on commands will not be available to batch users. Further restrictions on which commands and options are available to the batch user may prove desirable. For example, locating or altering source lines by matching text strings alone may lead to ambiguity which is readily apparent when editing interactively but may cause serious problems when entering corrections in batch mode.

5.1.6 UNIT

Some of the line editor commands use as a parameter the UNIT keyword, which has the short form UN. When this option is selected a text string will be considered a match with that given on the command only if it occurs as a unit, that is it is surrounded by characters other than those allowed in an SCL name. The most common example of this is to search for a character string that is enclosed in a pair of blanks.

5.1.7 VETO

Some of the line editor commands that involve text or source line alteration allow the keyword parameter VETO which has the short form V. If this parameter is specified, the first and last line of each group to be altered will be output followed by the query "CHANGE?", "DELETE?" or "REPLACE?" depending on the command. In the case of the CHANGE command the candidate line will be shown with the next string substitution already in effect. The user may make the following responses to the query:

NO 1 N = This causes the current group replacement to be 47 skipped. 48

11 2

1 3

1 4

: 5

1 6

1 7

8 9 10

11 12

13

14

15

16

17

18

19

20 21

22 23

24 25

26

27 28

29

30

31

32 33 34

35 36

37

138

139

:40

141

142

143

146

FRS	for	Source Code Utility (NOS 170 Version)	03/24/80 REVIJ
 5.0	SCU	EDITOR	, an
り。1 ~~~	•7 V	L I U 	ی این این این این این این این این این ای
		QUIT 1 Q 1 This causes the command further replacements.	to terminate with no
		CONTINUE 1 C : This causes veto mode current and any subsequent replacemen	
		YES 1 Y : This causes the current take place.	group replacement to
		Any other response causes the query t veto mode to remain in effect.	o be repeated and
	5.1.	8 END OF LINE BLANKS	
	sake	he SCU editor deletes trailing blank of matching character strings dur ar to be blank padded to "width" chara	ing searches lines
	5.1.	9 TERMINAL INTERRUPTS	· · · · · · · · · · · · · · · · · · ·
	have while the edite above QUIT inpu addi	nder IAF on 170 either user break 1 the same effect on the editor. If e displays are being sent to the termi output will be discarded and the edito command to be entered. If either or is waiting for a response to the ve e the editor will respond as if . If either is entered while the edit t, text input mode will be ended. tional carriage return will be require be able to continue.	either is received nal the remainder of or will wait for a is entered while the eto query described the user had entered or is accepting text In the last case an
	5.1.	10 REPLACEMENT TEXT	
	01		
		acement text may be supplied in three	
		cated by the use of a different keym processed regardless of the origin	
	text		
			consecutive sequence
		ers starting from the current value of this modification. The replacement	the sequence number text parameter is
	1 01	THE BOATITCALLAND THE LENIGGHGHT	ievi haiallei t2

							03/24/8	0
RS for	Source	Code Util	ity (NOS	5 170 Ve	rsion)	RE	EV: J	
.0 SCU	EDITOR							
•1•10 F	REPLACE	MENT TEXT						
	10 94 54 94 44 54 84 84	alle der Ann der der der Ann der Ann	alar shi sur din din din sur al	in the sin the sine the first the	na she dar dar she she she she she dar she o	the star star star the star star star st		~
descr	ribed b	elow:						
-								
[unti	il=diwl	th=t31file	= <local_< td=""><td>_file_na</td><td>ne>]</td><td></td><td></td><td></td></local_<>	_file_na	ne>]			
	until	t u t Wh	en the k	(eyword	UNTIL IS	used, rep	lacement	
		s read fro						
		ter strin						
		ies the de						
		ed. The d						
		the last c				delimite	er is not	
	consid	ered part	OT THE E	entered	TEXT.			
	with	1 w : W	han the	s with b	ovword ir	used. ++	a string	
		as the val				-		
	**	ement text				La angla		
			-					
	file	1 f = W	hen the	file ke	yword is a	used, rep	lacement	
	text i	s read fro	m the na	amed fil	e. If the	e keyword	FILE IS	
	used a	nd no valu	e is giv	ven an a	ttempt is	made to	access a	
		amed MERGE						
		can be us	ed as th	ne secon	d value o	f a value	e set for	
	This p	arameter.						
	r	ewind 1 r	1 rewind	d_before	1 rb I re	ewind_aft	er I ra	
	1	no_rewin	d 1 nr 1	These ke	ywords sp	ecify pos	sitioning	
	+	o be done	on the f	file. D	efault is	REWIND_E	BEFORE.	
	De f	ault value	s are as	signed	as follows	s. If	the the	
		of this						
		d to be a						
	omitte	d entirel	y the	command	will b	e process	sed as if	
	UNTIL=	*/* had be	en speci	ified.				
5.2	EDITOR	COMMANDS						
TI	he SCU	line edito	r commar	nds empl	ov a svi	ntax sin	nilar to	
		ed for NOS						
		Indicated,			-			
	-	two value			-			
		are optio					s within	
brack	kets a r	e an optio	nal part	t of an	optional	item.		
TI	he sho	rt form	for a	command	is given	with the	command	
where	e one i	s availabl	е.					
							N	

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J ~ ~ ~ ~ ~ ~ ~ ~ ~ 5.0 SCU EDITOR 5.2.1 CHANGE | C 5.2.1 CHANGE | C 1 2 This changes a specified number of occurrences of a 1 3 specified string to a replacement string. 1.4 5 change text=t1 [with=t3] [number=n] [lines=l1[..l2]] [unit] .. 1 6 [veto] 1 7 8 text 1 t : This specifies the text string which is to be 1 9 replaced on lines in the range given below. 110 11 with 1 w = This specifies the text string which will 12 replace the one searched for in the previous parameter. 13 If this parameter is omitted the original text string 14 will be deleted. 15 16 number 1 n : This specifies the maximum number of 17 occurences of the original string that are to be 118 replaced. The value of this parameter may be given as a 119 number or the keyword ALL. This defaults to ALL if a 120 range is given explicitly for the lines parameter and to 121 1 occurence when it is not. 22 23 124 lines | | : This specifies the range of lines to be scanned for string substitution. The value of 11 may be 125 given as a line identifier or one of the keywords ALL, 126 FIRST, CURRENT or LAST. The value of 12 may be given as 127 a line identifier or the keyword CURRENT or LAST. If 128 this parameter is omitted 11 and 12 are given the default 129 values of CURRENT and LAST respectively. 30 **31** unit 1 un = See description of unit above. 132 33 veto : v : See description of veto above. Also for this 34 command, should the string to be replaced occur more than 35 once in a line, the veto option will cause the line to be 36 displayed once for each potential replacement with the 37 next replacement in effect. 38 39 Example: change "Bill" "Sam" l=decka.153..LAST 149 41 The command above will replace every occurence of the text 142 string "Bill" with the text string "Sam" on each line in the 143 deck currently being edited from the line with the identifier 44 decka.153 to the last active line in the deck. 145 46 47 48

COMPANY PRIVATE

5

CUC SUPIWARE ENGINEERING STSTEM	03/24/80
ERS for Source Code Utility (NOS 170 Version)	REV: J
5.0 SCU EDITOR 5.2.2 CLEARTAB 1 CTAB	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
5.2.2 CLEARTAB 1 CTAB	
This clears tabbing selections defined by a T	AB command.
cleartab [columns=c1,c2cn]	
columns I col : This parameter is a list to be cleared. If this parameter is no columns are cleared.	
5.2.3 COPY	
This copies a group of lines from an libr file. The copied lines are left unaltered in location.	
<pre>copy [text=t1[t2]] [file=(<local_file_name></local_file_name></pre>	
[,rewind!rewind_before!rewind_after!no_rewind [lines=11[12]] [number=n] [unit]	nd])] ••
text 1 t : This specifies strings of tex	
the first and last lines of text to be copi- parameter is omitted entirely, the lines	
determine the lines to be copied. If a	
string is given, a group to be copied	
single line. When two text strings are give for the second starts immediately after the the first.	
file 1 f : This is the name of a local file	
text of the selected lines will be wri parameter is omitted the text lines will be	•
file named MERGE. The keywords below can	
second value of a value set for this parame	ter.
rewind 1 r 1 rewind_before 1 rb 1 rewi	nd_after 1 ra
1 no_rewind 1 nr :	
This parameter specifies positioning	
file. Omission causes the file to be it is written.	remound before
number In : This specifies the maximum nu	mber of groups
of lines to be copied. Number specifies	the count of
occurrences of the copied block when the	
is given and individual lines when it i parameter may have its value given as a	
keyword ALL. This parameter defaults to AL	
	COMPANY PRIVATE
	马口行把从马上 医式非脊椎毛足

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 5.0 SCU EDITOR 5.2.3 COPY is given explicitly for the lines parameter and to 1 when 11 It is not. 1 2 3 lines 1 1 : This specifies a range of lines to be scanned 1 6 for copying. I1 can be a line identifier or the keyword 1 5 ALL, FIRST, CURRENT or LAST. 12 may be given as a line 1 6 identifier or the keyword CURRENT or LAST. The default 1 7 values supplied if no range is given explicitly for the 1 8 lines parameter are CURRENT and LAST respectively. : 9 10 unit 1 un : See description of unit above. 111 12 Example: copy l=abc.4..abc.63 113 14 In this example the text lines abc.4 through abc.63 will be 115 copied out to a file called MERGE. 116 17 18 19 5.2.4 DECK 120 21 This command names the deck to which the following edit 122 commands are to apply. 123 24 125 deck name=deck_namelfirstlnextllast 26 127 name 1 na 1 first 1 next 1 last : When the keyword NAME or NA is used or a value only is given this specifies the 128 name of the deck to be edited. The alternative keywords 129 FIRST, NEXT and LAST can be used to select the decks 130 131 which occur in those positions in the directory on the 132 library. 33 Example: deck dsd 34 35 36 37 5.2.5 DELETE 1 D 38 39 This deletes specified lines from the current deck. 140 **4**Ť delete [text=t1[..t2]] [number=n] [lines=11[..l2]] [unit] .. 142 [veto] 143 44 text 1 t : This specifies strings of text occurring in 45 the first and last lines of text to be deleted. If this 46 parameter is omitted entirely, the lines parameter wilk 47 determine the lines to be deleted. If a single text 48

03/24/80

RS for Source Code Utility (NOS 170 Version)	REV: J
•0 SCU EDITOR •2•5 DELETE 1 D	
string is given, a group to be deleted wi single line. When two text strings are given for the second starts immediately after the c	n, the search
the first.	
number 1 n : This specifies the maximum number of lines to be deleted. Number specifies occurrences of matching blocks when the text given and individual lines when it is parameter may have its value given as a num	the count of parameter is not. This
keyword ALL. This parameter defaults to ALL is given explicitly for the lines parameter a it is not.	and to 1 when 1
lines 1 1 : This specifies a range of lines t for deleting. 11 can be a line identifier or ALL, FIRST, CURRENT or LAST. 12 may be giv identifier or the keyword CURRENT or LAST. values supplied if no range is given are CURR respectively.	r the keyword f ven as a line f The default f RENT and LAST f
unit : un : See description of unit above.	1
veto : v : See description of veto above.	4
Examples: delete "This line has a four letter word d	d." =a
The first example deletes all lines in the o which contain the indicated string. The sec deletes the current line.	current deck l cond example l l
5.2.6 DISPLAY_EDITOR_STATUS : DES	ť
This command displays the name of the file be the modification name in use, the deck being currently selected width, the current tab ch columns, what window columns are in effect and	g edited, the 1 haracter and 1 d whether the
verify and state options are active.	
display_editor_status	1

12

345

6 7

1 8

1 9

110

111

112

114

18

17

18

19

20

21

22

23 24

125

126

127

128

129

130

131

132

:33

34 35

136

137

138

139

140

141

142

143 44

145

03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 5.0 SCU EDITOR 5.2.7 END

5.2.7 END

This command terminates the line editor.

5.2.8 FIND 1 F

This finds a specified block of text in the current deck and updates the current position within the deck. If the verify option is selected the first and last lines in the block will be displayed or the line will be displayed if a single line is sought.

find [text=t1[...t2]] [occurrence=o] [lines=l1[...l2]] [unit]

text I t : This specifies strings of text occurring in the first and last lines of the block of text to be found. If this parameter is omitted entirely, the lines parameter will determine the line to be found. If a single text string is given, the block to be found will contain a single line. When two text strings are given, the search for the second starts immediately after the occurrence of the first.

occurrence 1 o : Occurrence specifies the count of occurrences of the matching block when the text parameter is given and single lines when it is not. Occurrence specifies that the oth occurrence of the line or block is to be found. The value of this parameter may be given as a number or the keyword LAST. The value of this parameter defaults to 1. When o is negative, the oth previous occurence of the line or block will be found starting with 11 from the lines parameter and scanning towards FIRST.

lines 1 1 : This specifies a range of lines to be scanned for the find. 11 may be given as a line identifier on the keyword ALL, FIRST, CURRENT or LAST. 12 may be given as a line identifier or the keyword CURRENT or LAST. 11 defaults to CURRENT and 12 defaults to LAST if the lines parameter is omitted entirely. If the value of the occurrence parameter is negative, 12 may not be given and the search is backwards from 11 to FIRST.

unit 1 un * See description of unit above.

Example: f *title* -1

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV:J 5.0 SCU EDITOR 5.2.8 FIND | F In this example the SCU editor will start from the current 1 1 line and then search for the nearest previous line which 2 includes the text string "title". 3 4 5 5.2.9 INSERT | I 6 7 This command inserts text after or before the specified 1 8 line. 1 9 10 insert {until=d1with=t3!file=<name>} [after=11!before=11] ... 111 [single_partition!multi_partition] 112 13 until I u I with I w I file I f : See discussion of 115 replacement text above. 115 16 after 1 af 1 before 1 be : This specifies the line text 117 is to be inserted after or before. This parameter may 118 have its value given as a line identifier or one of the 119 keyword FIRST, CURRENT or LAST. The default value for 120 this parameter is after CURRENT. 121 22 single_partition | sp | multi_partition | mp = This 123 parameter is only meaningful when the FILE keyword is 126 used to indicate the source of replacement text. If the 125 MULTI_PARTITION keyword is included on this command, each 126 end of record encountered in the merge file will be 127 represented on the source library as a *WEOP text 128 embedded directive. If the SINGLE_PARTITION keyword is 129 used or this parameter is omitted entirely, only the text 130 before the first separator on the source file will be 131 132 represented on the source library. 33 Example 1: insert af=bill.14 134 This line is inserted after bill line 14./ 135 36 Example 2: i "{Brief iname comment.}" 137 38 In this example the replacement text was given on the 139 148 command. 41 42 43 45 45

46

5-11

ERS fo	r Source Code Utility (NOS 170 Version)	03/24/80 REV:J
5.0 SC	U EDITOR LIST 1 L	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
** *** *** *** ***	** ** ** ** ** ** ** ** ** ** ** ** **	λα πο
5,2	•10 LIST 1 L	
dec	This lists specified blocks of lines k.	from the current
lis	t [text=t1[t2]] [number=n] [lines=l1[••12]] [unit]
	text 1 t : This specifies strings of the first and last lines of the bloc listed. If this parameter is omitted parameter will determine the lines to single text string is given, the bloc contain a single line. When two text the search for the second starts in occurrence of the first.	ck of text to be entirely, the lines be listed. If a ck to be listed will strings are given,
	number 1 n : This specifies the maximu of lines to be listed. Number specifies matching blocks when the text parame individual lines when it is not. This its value given as a number or the list parameter defaults to ALL when a range lines parameter and to 1 when no range	ecifies the count of eter is given and s parameter may have keyword ALL. This ange is given by the
	lines 1 1 3 This specifies the range listed. A value for 11 may be given a or one of the keywords ALL, FIRST, CU value for 12 may be given as a lin keyword CURRENT or LAST. If this para 11 defaults to CURRENT and 12 defaults	as a line identifier RRENT or LAST. A ne identifier or the ameter is omitted,
	unit t un : See description of unit al	bove.
	mple: 1 n=4 This lists 4 lines starting with the cur	rrent line.
5.2	-11 MASK	
35	This command defines a character which matching any character during string sea ection of such a character.	
	mask character=cloff	
	character 1 c 1 off # If the OFF keywo will be turned off. If the CHARACTER no keyword is used, this must be a c	keyword is used on

03/24/80

	03/24/80
S for Source Code Utility (NOS 170 Version)	REV: J
0 SCU EDITOR 2.11 MASK	
	ان میں میں بین میں میں بار کر ایک میں جو اور ایک میں کر ایک میں کر ایک میں میں میں ایک میں ایک ایک ایک ایک ایک ایک میں ایک
specifying the character which is to be characters in string searches.	used to match all
Example: Mask ***	
change "P*rr*t*n" "Perreten"	
In this example all vowels in the name	
These commands will correct all misspellings	
the current deck which match the pattern parameter.	given by the IEXI
5.2.12 REPLACE I R	
This deletes each group of lines in the	specified range
that starts and ends with lines containing the	
and replaces each group of lines with the rep	-
replace [text=t1[t2]] [until=d1with=t3]	file= <name>]</name>
	[veto]
[single_partition!multi_partition]	
text 1 t : This specifies strings of te	
the first and last lines of text to be re	· · · · · · · · · · · · · · · · · · ·
parameter is omitted entirely, the lines determine the lines to be replaced.	•
string is given, a group to be replaced	-
single line. When two text strings are	
for the second starts immediately after	the occurrence of
the first.	
until 1 u 1 with 1 w 1 file 1 f * 1	See discussion of
replacement text above.	dee AtgengstAH A₩
	,
number 1 n : This specifies the maximum (
of lines to be replaced. Number speci	
matching blocks when the text parameter Individual lines when it is not. This parameter	
its value given as a number or the key	-
parameter defaults to ALL when a	
explicitly for the lines parameter and to	
not.	
lines () This specifies a parse of 12	non to be nonned
lines : This specifies a range of lin for replacement. I1 can be a line id	
keyword ALL, FIRST, CURRENT or LAST.	
identifier or the keyword CURRENT or LAS	
values supplied if no range is give	n are CURRENT and

c

03/24/80

RS for	Source Code Utility (NOS 170 Version)	REV: J
.0 SCU	EDITOR	
	REPLACE 1 R	
	LAST.	
	unit I un #See description of unit above.	
	duri i du +266 describitou ol duri apore	
	veto t v t See description of veto above.	•
	single_partition 1 sp 1 multi_partition	n i mn i Thic
	parameter is only meaningful when the	
	used to indicate the source of replacement	nt text. If the
	MULTI_PARTITION keyword is included on the	
	end of record encountered in the merge	
	represented on the source library as embedded directive. If the SINGLE_PARTIN	
,	used or this parameter is omitted entire	•
	before the first separator on the source	-
	represented on the source library.	- - · · · -
Cwom.	ole: replace "call old" lines=ALL	·
CXGUL	call new(p1,p2,p3)/	
5.2.1	L3 SEQUENCE : SEQ	
T۲	nis command atters the form in which	h lines from the
	ary are displayed by the editor.	
seque	ence onloffiseparate	
	on 1 off 1 separate : If the ON keyword 1	is used on this
	command, the line identifier will precede	
	line on the display as part of the same i	
	keyword is used on this command the line	
	not be displayed. If the SEPARATE keywor line identifier will precede the tex	
	separate line. The initial setting of	
	taken from the value of the LID field in	
	deck when a deck is selected for editing	
	or a DECK command. If LID=NONE ther	N SEQUENCE OFF is
	selected, otherwise SEQUENCE ON is select	ted.
5.2.1	L4 STATE	
	This command causes the state of	the modification
	associated with the introduction of a	
	displayed when that line is displayed.	
	state onloff	

5-15 03/24/80

1 2

1 3

14

1 5

1 6

1 7

1 8

: 9

110

112

114

115

116

117

:19

120

21

122

123

124

125

27

28 29 38

31 32

133

134

135

136

38

39 40 41

142

ERS for Source Code Utility (NOS 170 Version) REVIJ 5.0 SCU EDITOR 5.2.15 TAB 1 T

5.2.15 TAB | T

This command alters the tabbing environment for the INSERT and REPLACE commands. The user specifies a tab character and columns. The presence of the tab character in input text causes blank fill to the next tab column. Tab characters entered after the last tab column has been reached will be accepted as data. Use of the DECK command causes the default tab selections for the named deck to be invoked.

tab [character=c] [columns=c1,c2,...cn]

character 1 c : This one character string specifies the tab character. If this parameter is omitted, the currently selected tab character will remain in effect.

columns 1 col : This gives tab columns to be selected. A maximum of 256 tab columns may be selected at any time. These integers must be in the range 1 to 256. Multiple columns may be specified as a value list (see the Parameter Lists section of the NOS/VE ERS). If this parameter is omitted entirely the currently selected tab columns will remain in effect.

Example: tab "\" (11,18,30,36)

5.2.16 VERIFY # V

This command causes the CHANGE command to echo back to the user the altered line(s). In addition if the verify option is selected the first and last lines of a block located by the FIND command will be displayed or the line will be displayed if a single line is sought.

verify ontoff

5.2.17 WIDTH 1 W

This command declares the number of characters the editor 144 will allow to be entered in a text line without issuing a 145 warning message. Use of the DECK command causes the default 146 width from the deck header to be invoked. 147

ERS for Source Code Utility (NOS 170 Version)	03/24/80 REV:J
5.0 SCU EDITOR 5.2.17 WIDTH : W	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
width width=n	
width 1 w : The value for this parameter an integer from 0 to 256. Specifying 0 of up to 256 characters to be entered.	
5.2.18 WINDOW 1 WN	
This command limits the range of character a line to be scanned in string searches editor commands. Only the characters in the searched for a match with the string give DELETE, LIST, REPLACE, FIND, COPY and CHANGE.	during subsequent given range are
window [columns=c1[c2]]	
columns 1 col : This specifies the star scan columns. c1 defaults to column 1 256.	

	03/24/80
S for Source Code Utility (NOS 170 Version)	REV: J
O SCU EDITOR	
3 EDITOR FUNCTIONS	هنه هم چې چې چې د وره چې د ده ده چې ده چې ده چې چې د ور چې چې دې چې چې چې د
5.3 EDITOR_FUNCTIONS	
Specialized command language functions wi	
within the version of the SCU editor which	
NOS/VE. They are intended to be used in conju System Command Language control statements to	
the path taken through sequences of SCU	
These functions will not be available in the	
5.3.1 \$CURRENT_DECK	
This function will return the name of	f the currently
selected deck as its value.	
5.3.2 SFIRST_DECK	
This function will return the name of the (deck which occurs
first in the directory of the library as its w	
5.3.3 \$LAST_DECK	
Jeded DEAGT_DEOR	
This function will return the name of the (deck which occurs
last in the directory of the library as its va	atue.
5.4 EDITOR LOOPING	
XAA FRETCH-FAATFIC	
On NOS/VE it will be possible to com	
commands and functions with SCL control states	Ŧ
	ecssary in such
applications as test base maintenance.	

- 43 44 45
- 46

۹.

1234567

8

9

10

11 12

13 14 15

116

118

119

20 21

22 23

124

126

127

28

29 30 31

132

134

135

40

41

42

143

146

145

03/24/80

ERS for Source Code Utility (NOS 170 Version) REV:J 6.0 FEATURES UNDER CONSIDERATION

6.0 FEATURES UNDER CONSIDERATION

The features listed in this section are under consideration. Some are listed here because it is felt that they may prove superfluous. In other cases they are listed here because it is questionable whether they are compatible with the basic design of the utility or it is not yet clear in what form the feature should be supplied.

6.1 MODNAME EDITOR COMMAND

It may prove desirable to provide a editor command similar to the MODNAME directive in MODIFY, which would establish a default modification name allowing a shorthand form for line identifiers.

6.2 INACTIVE LINES AND THE EDITOR

Currently it is planned that the SCU editor deal with only active lines. It may prove desirable for the editor to FIND or LIST inactive lines or perform other manipulations on inactive lines.

6.3 <u>RESTORE LINE COMMAND</u>

It may prove desirable to provide a RESTORE command in the editor, which would clear the inactive flag for lines previously deleted.

6.4 CHANGES NOT INTRODUCED IN THE EDITOR

Some SCU commands outside the editor introduce significant changes to a source library without being associated with a modification name. No effective method for automating the control of this type of change has been devised. Those commands of concern in this respect are PURGE_DECK, SEQUENCE_DECK, PURGE_MODIFICATION and SEQUENCE_MODIFICATION.

⁴⁶ 47 48

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 6.0 FEATURES UNDER CONSIDERATION 6.5 AUTOMATIC MODIFICATION RESEQUENCING 6.5 AUTOMATIC MODIFICATION RESEQUENCING 1 2 may be desirable to automatically resequence It 3 modifications when their state is raised to 1 4 (developmental). See the SEQUENCE_MODIFICATION command. 5 6 7 6.6 SAVING EDIT FILES 8 9 It may be desirable to provide a mechanism for preserving 119 an library as permanent without exiting the editor. 111 12 13 6.7 KEY CHARACTER_SUBSTITUTION 14 15 It may be desirable to allow the CHANGE LIBRARY command to 16 substitute the key character used on a source library. 117 18 19 6.8 COMPARE UTILITY 120 21 It may prove useful to supply a utility which could compare 122 a source file with a deck on an SCU library and produce a set 123 of editor commands to cause the deck to match the source 124 file. 125 26 27 6.9 PROCESSOR BASED DECK SELECTION 128 29 prove useful to provide a capability to T† mav 138 conditionally write a deck to an expanded text file based on 131 the contents of the processor field in the deck header. 132 33 34 6.10 EDITOR MOVE COMMAND 135 36 It may prove useful to provide an editor command to move a 137 block of text lines from one place in a deck to another. 138 39 48 41 42 43 44 45 46 47 48

COMPANY PRIVATE

				Version)	REV: J	
	SED IMP					

-

7.0 PHASED IMPLEMENTATION

Current plans are to implement the version of SCU that executes under NOS 170 in the steps below. Design of the version of SCU to execute on NOS/VE has begun and implementation will tentatively begin in March of 1980.

- All SCU features will be implemented and tested by 1. the project by April, 1980.
- 2. SCU will be ready for release as part of the SES tools package, as part of SES Release 14, currently scheduled for July, 1980.

1 7

1 8

1 9

:12

12345678

; 9 10

11T

112

113

114

115

116

124

129

36

42 143 44

47

148

		-		Version)	REV: J	

8.0 MESSAGES

8.0 MESSAGES

8.1 NON-EDITOR MESSAGES

Below are listed messages issued primarily from non-editor SCU procedures. The conditon numbers are shown relative to 9000. This origin may change on 170 and will change on NOS/VE. The letters listed in the severity column stand for E - error, F - fatal, I - informative and W - warning respectively. Where the characters +T are shown below some string will be substituted in the actual message.

Condition	Severity_	<u>Message</u>		
9001	E	file +T	not	local

9002 E file +T is not SCU libra	9002	Ε	file +T is	not	SCU	librar
---------------------------------	------	---	------------	-----	-----	--------

The contents of the file +T are not recognizable by the utility as an SCU source library.

9003 E em	pty SCU library - +T
-----------	----------------------

There are no decks on the SCU library on file +T.

			30
9004	Ε	deck +T not found	131
			32
9005	Ε	mod +T not found	133
			34
9006	E	conflict in file name - +T	135

An Example of when this message is issued is naming the 137 same file for two different parameters on a command such as 138 EXM MODNAME BA=FILE EC=FILE. The message will also be issued 139 if the user attempts an operation that would be destructive to 140 the result file named on the SCU.GROUP command. 141

07 E duplicate deck name – +T

The user has mentioned the same deck twice on the same 145 command. 146

9008 E duplicate mod name – +T

03/26/80

for Sourc	e Code Utili	ty (NOS 170 Version)	03/24/80 REVIJ
MESSAGES		, ago ago dan ina ang ana ang ang ang ang ang ang ang a	ay nu da su ay nu nu nu nu nu nu nu na na ha ha ha ha ha ha ha ha ha
NUN-EUITU	R MESSAGES	و من من هذه من	an a
The use	r has attemp	ted to create a modification	on with the
		dy existing on the source l	
9009	Ε	deck +T already on library	
The up	on has attam	pted to create a deck with	the come name
		ic on the source library.	THE Same Hame
9010	E	emoty deck - +T	
No text	lines have	ever been introduced into d	eck +T.
9011	E	invalid deck range - +T to	+ T
One of	the deck nam	es given as one of the ends	of a range
		the library or they are no	t in order of
occurrence	on the libr	9L A •	
9012	E	invalid mod range - +T to	+T -
One of	the modifica	tion names given as one of	the ends of a
		on the library or they are	
		library. Displaying the hould suggest which.	modification
TIST TOP T	he ithrary 2	nouru suggest warca.	
9013	£	invalid parameter value fo	r +T
An inv	atid value	has been given for parame	ter +T on the
		urrently being scanned.	
9014	E	unexpected +T encountered	
3014	<u> </u>		
		text embedded directive (E	-
	which has being proces	been encountered in the tex sed.	T OT THE DECK
9015	E	expected +T not found	
SCU has	searched to	the end of a deck without	encountering
a matching	text embedd	led directive (IFEND or TEXT	END).
9016	ε	user without proper author	ity
		empted to perform some comm the user names assigned auth	
library.		The Aser Hames assigned duit	
•			

split.

03/24/80

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~		Utility (NOS 170 Version)	ین میں میں بین بین میں اس میں بند میں میں بین میں بین میں میں میں میں میں میں ا
MESSAG	ES ITOR MESSA	GES	
یوی بری بری بری بین بین بری بری	~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~		राज तेवर तेवर तेवर तेवर तिव तिव तेवर त्यां ^{राज} तिव व्यक्त क्या का प्राप्त का त्या त्या त्या का त्या त्या का
9017	Ε	deck +T already interloc	ked
9018	Ε	illegal directive +T	
	*	n given on the displayed *I recognizable.	F text embedded
9 019	Ε	too many nested IF*s	
		ested his IF directives bey ed by SCU (32).	ond the depth
9020	Ε	too many alternate bases	
	•	allows the specification e commands that allow multiple	
9021	е Е е с	+T value too large	
+T i too lar	-	ord for the parameter which wa	s given a value
9022	Ε	mod +T not state 4	
9023	E	mod +T state 4	
9024	Ε	+T	
		of this message the entire e	rror message is
9025	Ε	file +T without write pe	rmission
9026	E	grouping commands. BA RESULT must be specified	
9027	E	feature +T not found	
9028	E	group +T not found	
9029	Ε	different key character	on file - +T
9030	I	line +T in deck +T trunc	ated
The an EXPA		line in the indicated deck w EXTRACT_DECK command. A larg	

have to be supplied on the command or the line will have to be

COMPANY PRIVATE

147

13/26/80

NON-ED	ES ETOR MESSAG	SES	
0031	Ε	interlock violation in de	eck +T
	the proces	olation has occurred for the ising of a COMBINE_LIBRARY or I	
090	F	unable to allocate +T	
_mod_na o a dec	ames, used :k. Sh <u>o</u> rt	tables allocated in memo for the list of modification of restructuring his librar an do to correct this problem.	names applying
099	F	internal error +T	
ihere i	is no rem	lght be a library pointer of medial action available to a documenting the problem so that	the user beyond
		can be sought out and correct	
nherever		e can be sought out and correct	
Belo Belo Belo Drocedur 1000- 105/VE- 10110wir	It may be OR MESSAGE are lis res. The This orig The lette ng signifo warning.	e can be sought out and correct	the SCU editor n relative to will change on lumn have the E - informative
below Below Below Boo- Boo- Boo- Boo- Boo- Boo- Boo- Bo	It may be OR MESSAGE are lis res. The This orig The lette ng signifo warning.	E can be sought out and correct Sted the messages issued by condition numbers are shown gin may change on NOS 170 and ers given in the severity co cance E - error, F - fatal, The characters +T will be actual messages.	the SCU editor n relative to will change on lumn have the E - informative
Below Below Below Below Boo- Boo- Boo- Boo- Boo- Boo- Boo- Bo	It may be <u>IOR MESSAGE</u> are listes. The This origon The letten ag signifor warning. ext in the <u>E</u>	E can be sought out and correct Sted the messages issued by condition numbers are shown gin may change on NOS 170 and ers given in the severity co cance E - error, F - fatal, 1 The characters +T will be actual messages. CYMessage	the SCU editor n relative to will change on lumn have the I – informative replaced with
berever Below Below Procedur 1000- 100/VE- 1001/VE- 1001 The 1001	It may be <u>IOR MESSAGE</u> are listes. The This orig The letten g signifor warning. ext in the <u>Severit</u> E string of	E can be sought out and correct Sted the messages issued by condition numbers are shown gin may change on NOS 170 and ers given in the severity co- cance E - error, F - fatal, The characters +T will be actual messages. Ty Message Unknown command: +T	the SCU editor n relative to will change on lumn have the I – informative replaced with
A condition 3002 The a	It may be IOR MESSAGE are listers. The This origon The letter ag signifor warning. ext in the <u>on Severit</u> E string of command. E user has no	E can be sought out and correct Sted the messages issued by condition numbers are shown ain may change on NOS 170 and ers given in the severity co cance E - error, F - fatal, The characters +T will be actual messages. CY Message Unknown command: +T characters +T was not recognized	the SCU editor n relative to will change on lumn have the I - informative replaced with

COMPANY PRIVATE

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REV: J 8.0 MESSAGES 8.2 EDITOR MESSAGES 8004 E Continue specified for unknown 11 modification +T 1 2 3 The user has specified continue on the editor call command 1 4 and the modification +T does not exist on the library. 1 5 1 6 Issuing a DISPLAY_MODIFICATION_LIST command for the library may point out a misspelling. 1 7 8 8005 Ε Duplicate modification name +T 1 9 10 The user has called the editor naming a modification name 111 which already exists on the library without specifying the 112 CONTINUE keyword. He must either choose a different 113 modification name or specify the keyword CONTINUE to add to 114 the named modification. 115 16 8006 Ε Modification +T not in state 0 117 18 SCU does not allow modifications in states other than 0 to 119 be changed. 120 21 8007 E Invalid Modification name 122 23 E 8008 Invalid keyword value for NUMBER - +T 124 25 The only valid keyword value for the number parameter is a 126 ALL or A. 127 28 8009 E Value for NUMBER must be positive 129 integer or keyword 130 31 8010 E Invalid file name 132 33 8011 Ε Invalid file position specified 134 35 Valid keywords for file positioning are REWIND or R. 136 REWIND_BEFORE or RB, REWIND_AFTER or RA and NO_REWIND or NR. 137 38 8012 E Column number too large 139 40 Values for the COLUMNS parameter must be between 1 and 141 256. 142 43 8013 E Width greater than maximum line length 144 45 Values for width must be between 0 and 256. 146 47 8014 E Width greater than maximum width for 148

CDC SOFTWARE ENGINEERING SYSTEM 03/24/80 ERS for Source Code Utility (NOS 170 Version) REVIJ 8.0 MESSAGES 8.2 EDITOR MESSAGES ------deck The user has attempted to give a value for width greater than that in the deck header on the library. Modification name +T too long 8015 Ε 8016 Ε Line identifier with must start modification name 8017 Ε Sequence number too large The largest value SCU allows for a sequence number is 262143. 8018 Ε Sequence number required after modification name 8019 Ε Invalid keyword value +T An invalid keyword value has been given for the LINES or

AFTER/BEFORE parameter. 8021 Ε Keyword ALL cannot be used in a range

	· · · · ·		25
8022	E	Range of lines cannot be specified for	126
		a backward search	127
			28

Unknown modification name +T

The SCU editor does not recognize +T as the name of a modifcation on the library.

8024 Ε Line at end of range is inactive

The editor operates on active lines only.

8023

Ε

8025	E	Last line in range not found after	138
000	·	first line	139
			413
0077	~	First line is seen ast found before	· •
8027	E	First line in range not found before	141
		last line	142
			43
8028	E	First line in range not found	144
			45
8029	Ε	First line in range not active	146
		-	47
The	editor ope	erates on active lines only.	148

1 1 2

1 3

1 6 7

1 8

1 9 11

111 12

113

114 15

116

117 18

119 28

121

122 23

124

129 38

131

132 33

134

35

136 37

MESCAC	50	
MESSAG	MESSAGES	
يناد نربد فإناء جال والا سنة	ا حله دیار عله اعله وی علی اعلی بید عله علم .	
030	E	Text string +T not found
031	I	+T occurrences found
This	message	informs the user how many occurrences of a
		ound if the user asked to find all or some
		than existed in the range of lines to be
earche	d.•	
032	E	No active lines in range
***	aditos	nator on onting liner onto
ine	editor ope	rates on active lines only.
033	I	+T lines to end of range
This	maccana 1	nforms the user how many lines were found in
	-	the range of lines to be accessed if the user
		on all lines or some number greater than
xist i	n the rema	inder of the range.
034	W	Line is longer than maximum line length
		Line is longer than maximum line length es the user has introduced was greater than
One 56 ch	of the lin aracters	
One 56 ch	of the lin aracters	es the user has introduced was greater than
One 56 ch haract	of the lin aracters	es the user has introduced was greater than
One 56 ch haract 035	of the lin aracters ers. I	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T
One 56 ch haract 035	of the lin aracters ers.	es the user has introduced was greater than in length. It has been truncated to 256
One 56 ch haract 035 036 The	of the lin aracters ers. I E structure	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no
One 56 ch haract 035 036 The	of the lin aracters ers. I E structure	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck
One 56 ch haract 035 036 The emedia	of the lin aracters ers. I E structure	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no
One 56 ch haract 035 036 The emedia 037	of the lin aracters ers. I E structure 1 action a E	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user. History array for line has overflowed
One 56 ch haract 035 036 The emedia 037 The	of the lin aracters ers. I E structure 1 action a E	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user.
One 56 ch haract 035 036 The emedia 037 The	of the lin aracters ers. I E structure 1 action a E	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user. History array for line has overflowed
One 56 ch haract 035 036 The emedia 037 The imes.	of the lin aracters ers. I E structure 1 action a E line in	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user. History array for line has overflowed
One 56 ch haract 035 036 The emedia 037 The imes. 038	of the lin aracters ers. I E structure 1 action a E line in W	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user. History array for line has overflowed question has been modified more than 255 Line is longer than current line width
One 56 ch haract 035 036 The emedia 037 The imes. 038 This	of the lin aracters ers. I E structure 1 action a E line in W is a warn	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user. History array for line has overflowed question has been modified more than 255 Line is longer than current line width ing that the user has introduced a line
One 56 ch haract 035 036 The emedia 037 The imes. 038 This onger	of the lin aracters ers. I E structure 1 action a E line in W is a warn than the	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user. History array for line has overflowed question has been modified more than 255 Line is longer than current line width
56 ch haract 035 036 The emedia 037 The imes. 038 This onger	of the lin aracters ers. I E structure 1 action a E line in W is a warn than the	es the user has introduced was greater than in length. It has been truncated to 256 Begin editing deck +T No start of deck of the library is not intact. There is no vailable to the user. History array for line has overflowed question has been modified more than 255 Line is longer than current line width ing that the user has introduced a line current line width and should consider

D MESS 2 EDI1	ESSAGE	ES			م جد جاد چې جان چان کې			ور بناه شد چه چه خد بنه ب	a dag saga sing sing sing sing s
		accepta AST or	ble key L•	word va	atue	for	the	OCCURRE	INCE
8040	E		Value keywor	for occu d	arrenc	e mus	t be	integer	
					-				
,									
	•								
							CONT	PANY PRI	

03/24/80

9-1

ERS for Source Code Utility (NOS 170 Version) REV:J 9.0 SUGGESTIONS ON SCU USE

9.0 SUGGESTIONS ON SCU USE

The degree of control necessary over corrections to source data blocks being maintained for an individual for his own convenience and over the source code for a major operating system component being maintained by an integration and evaluation group is very different. For that reason the two groups of suggestions are offered below.

9.1 PRIVATE SOURCE CODE MAINTENANCE

For the convenience of an individual maintaining a private source library it is suggested that he set all authorities to his own user name. Probably the only criteria file directive that he would need to use would be DECLARE, which is useful in allowing alternate blocks of source code to be written to an expanded text file.

9.2 GROUP SOURCE CODE MAINTENANCE

It is suggested that for maintenance of complex blocks of source data that may be required to be accessible to a large group of implementors that the authority and interlock features of SCU be used to ensure control over modifications being made. It is suggested that a user who is assigned the responsibility to implement a correction to a particular deck extract a subset library containing only the deck that he is to modify, setting the interlock for that deck in the process.

The individual could then enter his correction and make trial use of it from his subset source library satisfying needs for other decks from the base library specified as an alternate. The individual would finally submit his modified subset library for code review and combining with the base library with the new corrections in state 0. The interlock would be cleared when the combine was done.

As various stages of testing are completed the person in 45 authority can grant that the new corrections be raised in 47 state or if testing indicates that the correction is not 48

45678

9

110

111

112

123

17

18

19

20

21

22 23

24 25

26 27

28

29

30

31

32

33

34

35 36

37

138

139

40

41

42

145

03/24/80

9-2

							Version)	REV:J	,
9.0 9.2	SUGO	GESTIONS	S ON S Ce coi	SCU De i	USE MAIN	ENANCE		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	

acceptable can lower the state of the correction to state 0 and contact the person responsible to make an adjustment before resubmitting the correction.

Expanded text files can be built with all available corrections on the base source library to check for conflicts between developers or at the other extreme can be built with only released code to produce a version of the source data at some significant milestone.

The DISPLAY_LIBRARY, DISPLAY_DECK, DISPLAY_MODIFICATION, DISPLAY_GROUP and similar commands can be used to help document and evaluate subset source libraries prior to combining them with the base source library.

Criteria files can be used to specify the contents of a complex expanded text file. The contents of these files can of course be maintained on the source library themselves.

Ĩ

1 03/24/80

	Table of	Cont	ents								1
											2 3
											9 5
											5
1.0 PREFACE										1-1	-
1.1 SCOPE OF DOCUMENT .											7
1.2 APPLICABLE DOCUMENTS										1-1	8
ISE AFFEIGABLE BOGONENTS		• •	* * *	• •	• •	9 99	• •	•	• •	1-1	9
2.0 INTRODUCTION										2-1	10
2.1 CONCEPTS											
2.1.1 SOURCE LIBRARY .											12
2.1.2 SOURCE DECK										2-2	
2.1.3 GROUP											
2.1.4 MODIFICATION											15
2.1.5 FEATURE											16
2.1.6 STATES											17
2.1.7 AUTHORITY											18
2.1.8 CRITERIA FILE .											19
2.1.9 TEXT EMBEDDED DI	RECTIVES	• •	• • •	• •				· •	• •	2-4	20
2.1.10 NORMAL SEARCH O	RDER			• •	• •	•				2-4	21
2.1.11 BASIC STATUS CO											22
2.1.12 NAMES										2-5	23
2.1.13 STRINGS		• •		• •		•	• •	•	• •	2-5	24
2.1.14 LOCAL FILES		• •		• •		•		٠		2-5	25
											26
3.0 COMMANDS											27
3.1 SOURCE LIBRARIES											28
3.1.1 CREATE_LIBRARY :										3-2	29
3.1.2 DISPLAY_LIBRARY										3-4	30
3.1.3 CHANGE_LIBRARY 1										3-5	31
3.1.4 COMBINE_LIBRARY										3-7	32
3.1.5 ADD_LIBRARY 1 AD										• •	33
3.1.6 REPLACE_LIBRARY										3-11	34
3.1.7 EXTRACT_LIBRARY										3-13	35
3.2 SOURCE DECKS										3-15	36
3-2-1 CREATE_DECK I CR											37
3.2.2 DISPLAY_DECK D 3.2.3 CHANGE_DECK CH		• •	• • •	• •	• •	•	• •	٠	• •	3-20	38
										3-22	39
3.2.4 PURGE_DECK 1 PUD 3.2.5 SEQUENCE_DECK 1										3-25	40
3.2.6 MOVE_DECK : MOD										3-26 3-27	41 42
3.2.7 EXPAND_DECK 1 EX											42
3.2.8 EXTRACT_DECK 1 E										3-32	43 44
3.2.9 DISPLAY_DECK_REF										3-35	45
3.2.10 DISPLAY_DECK_LI										3-35	46
3.2.11 DISPLAY_GROUP 1			·			-		-		3-37	47
3.2.12 DISPLAY_GROUP_L	TST I DGI		·			-		-		3-37	48
3.3 MODIFICATIONS		• •	• • •	• •				•	• •	3-39	49
3.3.1 CREATE_MODIFICAT	ION 1 CRI	ч <u>-</u>	• •	• •		-		-	• •	3-39	50
3.3.2 DISPLAY_MODIFICA										3-40	51
3.3.3 CHANGE_MODIFICAT										3-42	52
3.3.4 PURGE_MODIFICATI	ON 1 PUM	• •	• •	• •		•		•	• •	3-44	53
3.3.5 SEQUENCE_MODIFIC	ATION 1 S	SEM		• •			• •		• •	3-45	54
					-	-	-				

03/24/80

3.3.6 EXTRACT_MODIFICATION : EXM • • • • • • • • • • • • • • • • • • •	3-46	1
3.3.7 DISPLAY_MODIFICATION_LIST DML	3-47	2
3.3.8 DISPLAY_FEATURE 1 DIF	3-47	3
	3-48	4
3.4 CONVERSION AIDS	3-49	5
3.4.1 CONVERT_UPDATE_TO_SCU : CUTS	3-50	6
3.4.2 CONVERT_MODIFY_TO_SCU 1 CMTS	3-51	7
3.4.3 CONVERT_SCU_TO_UPDATE 1 CSTU	3-52	8
3.5 CALLING THE EDITOR	3-54	9
3.5.1 EDIT	3-54	18
3.6 GROUPING SCU COMMANDS	3-56	11
3.6.1 GROUP	3-56	12
3.6.2 END	3-57	13
	0 21	14
4.0 DIRECTIVES	4-1	15
4.1 TEXT EMBEDDED DIRECTIVES	4-1	16
4.1.1 COPY	4-1	17
4.1.2 COPYC (COPY CONDITIONAL)	4-2	18
4.1.3 DECK	4-2	19
4.1.4 ELSE	4-2	21
4.1.5 IF • • • • • • • • • • • • • • • • • •	4-2	21
4-1-6 IFEND	4-5	22
4.1.7 TEXT	4-5	23
4.1.8 TEXTEND	4-5	25
	4-5	
		25
4.1.10 WEOPC	4-5	26
4.2 CRITERIA FILE DIRECTIVES	4-6	27
4.2.1 DECLARE	4-6	28
4-2-2 EXCLUDE_DECK	4-6	29
4.2.3 EXCLUDE_FEATURE	4-6	30
4.2.4 EXCLUDE_GROUP • • • • • • • • • • • • • • • • • • •	4-7	31
4.2.5 EXCLUDE_MOD	4-7	32
4.2.6 EXCLUDE_STATE	4-8	33
4.2.7 INCLUDE_COPYING_DECKS	4-8	35
4.2.8 INCLUDE_DECK	4-8	35
4.2.9 INCLUDE_FEATURE	4-8	36
4.2.10 INCLUDE_GROUP	4-9	37
4.2.11 INCLUDE_MOD	4-9	38
4.2.12 INCLUDE_STATE	4-10	39
		40
5.0 SCU EDITOR	5-1	4 1
5.1 EDITOR CONCEPTS	5-1	42
5.1.1 TABBING	5-1	43
5.1.2 LINE IDENTIFIERS	5-1	44
5.1.3 KEYWORD VALUES	5-2	45
5.1.4 WIDTH	5-3	46
5.1.5 THE BATCH USER	5-3	47
5.1.6 UNIT	5-3	48
5.1.7 VETO	5-3	49
5.1.8 END OF LINE BLANKS	5-4	50
5.1.9 TERMINAL INTERRUPTS	5-4	51
5.1.10 REPLACEMENT TEXT	5-4	52
5.2 EDITOR COMMANDS	5-5	53
5.2.1 CHANGE C	5-6	54
	-	•

03/24/80

5.2.2 CLEARTAB CTAB	• 5-7 t
5.2.3 COPY	
5.2.4 DECK • • • • • • • • • • • • • • • • • • •	
5.2.5 DELETE 1 D	• 5-8 4
5.2.6 DISPLAY_EDITOR_STATUS 1 DES	
5.2.7 END	• 5-10 5
5.2.8 FIND 1 F	
5.2.9 INSERT ! I	
5.2.10 LIST I L	
5.2.11 MASK	• 5-12 10
5.2.12 REPLACE R	
5.2.13 SEQUENCE I SEQ	
5.2.14 STATE	• 5-14 13
5.2.15 TAB T	
5.2.16 VERIFY V	
5.2.17 WIDTH I W	
5.2.18 WINDOW WN	
5.3 EDITOR FUNCTIONS	
5.3.1 \$CURRENT_DECK	
5.3.2 \$FIRST_DECK	
5.3.3 \$LAST_DECK	
5.4 EDITOR LOOPING	• 5-17 22
	23
6.0 FEATURES UNDER CONSIDERATION	• 6-1 24
6.1 MODNAME EDITOR COMMAND	• 6-1 25
6.2 INACTIVE LINES AND THE EDITOR	• 6-1 26
6.3 RESTORE LINE COMMAND	• 6-1 27
6.4 CHANGES NOT INTRODUCED IN THE EDITOR	
6.5 AUTOMATIC MODIFICATION RESEQUENCING	
6.6 SAVING EDIT FILES	• 6-2 38
6.7 KEY CHARACTER SUBSTITUTION	• 6-2 31
6.8 COMPARE UTILITY	• 6-2 32
6.9 PROCESSOR BASED DECK SELECTION	
6.10 EDITOR MOVE COMMAND	• 6-2 34
	35
7.0 PHASED IMPLEMENTATION	. 7-1 36
	37
8.0 MESSAGES	. 8-1 38
8.1 NON-EDITOR MESSAGES	• 8-1 39
8.2 EDITOR MESSAGES	• 8-4 40
	41
9.0 SUGGESTIONS ON SCU USE	• 9-1 42
9.1 PRIVATE SOURCE CODE MAINTENANCE	• 9-1 43
9.2 GROUP SOURCE CODE MAINTENANCE	• 9-1 44