

Consiglio Nazionale delle Ricerche

Common Command Language

Support on STAIRS/VS - TLS

Implementation Description

R. Bartoli - S. Lippi - G. A. Romano - O. Signore

177

GNUCE

A cura di: R. Bartoli (CNUCE)
S. Lippi (IBM, Italy)
G. A. Romano (CNUCE)
O. Signore (CNUCE)

Copyright - Ottobre 1980

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

DDDDDDDDDD		BBBBBBBBBBBB
DDDDDDDDDD		BBBBBBBBBBBB
DD	DD	BB BB
DD	DD	BB BB
DD	DD	BB BB
DD	DD	BBBBBBBBBB
DD	DD	BBBBBBBBBB
DD	DD	BB BB
DD	DD	BB BB
DD	DD	BB BB
DDDDDDDDDD		BBBBBBBBBBBB
DDDDDDDDDD		BBBBBBBBBBBB

Common Command Language
 Support on STAIRS/VS - TLS
 Implementation Description

Reparto
 Basi di dati e sistemi informativi

R. Bartoli (CNUCE)
 S. Lippi (IBM, Italy)
 G.A. Romano (CNUCE)
 O. Signore (CNUCE)

CNUCE
 Via S. Maria, 36
 56100 Pisa

Tel. +39 50 45245
 Telex 500371 CNUCE

October 1980

Table of contents

Foreword	1
Installation guidance	2
A. SCSTWA	10
B. Correspondence table	16
C. Implementation Logic	18
D. General support modifications	19
D1. TTY support	20
D2. Input conversion	21
D3. Conversational/nonconversational input	22
D4. CCL command definition	22
D5. SHORT/LONG dialog	23
D6. Message file selection	23
D7. Map selection	24
E. Command Implementation	25
E1. Sign on	26
E2. HELP Command	29
E3. BACK Command	30
E4. MORE Command	31
E5. PAGE Command	32
E6. DEFINE Command	33
E7. STOP Command	35
E8. OWN Command	36
E9. BASE Command	37
E10. DISPLAY Command	39
E11. FIND Command	42
E12. SHOW Command	48
E13. PRINT Command	50
Appendix A: Field labels	52

Foreword

The EURONET CCL (Common Command Language) was devised to overcome the problems which may arise on encountering unfamiliar commands when using more than one retrieval system.

As many of the functions provided by the different information retrieval systems available on the market are similar, these shared functions have been standardised in the CCL.

A contract was stipulated between the EEC and CNUCE for the implementation of CCL on STAIRS/VS.

This publication describes the implementation made under the terms of this contract which is already distributed to the EURONET hosts using STAIRS/VS - TLS.

The reader is supposed to be familiar with the CICS, STAIRS/VS and TLS environments.

Installation guidance

A copy of this report is contained in the first file of the distribution tape.

The following JCL is suggested to obtain a print-out of the tape:

```
//NOTICE JOB
//      EXEC   PGM=IEBGENER
//SYSPRINT DD  SYSOUT=A
//SYSIN   DD   DUMMY
//SYSUT1  DD   DSN=SCSMEMO,UNIT=TPV9,VOL=SER=GNUCE,
//          DISP=OLD,LABEL=(1,SL),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSUT2  DD   SYSOUT=A,DCB=RECFM=FBA,LRECL=132,BLKSIZE=1320)
```

The text is written in upper and lower case characters.

The distribution tape is 9 track, 1600 BPI, labeled GNUCE. Upon request, an unlabeled tape can be distributed. The JCL suggested above refers to the standard labeled tape, if an NL tape is used, the JCL must be changed.

The tape contains 11 files:

1. This note
2. SCS.TEST.MACLIB MACLIB to compile SCS modules, which contains:

DLNSTART	modified DLNSTART macro
SCSCMDEF	macro for CCL Commands Table (SCSCMDEF) generation
SCSTWA	CCL Transaction Work Area (Assembler)
TSDPLYRD	Temporary Storage Display Record Descriptor Dummy Section
TSDPLYR0	Temporary Storage Display Record 0 DSECT
TS047RD	DSECT for Temporary Storage Record written by DLN047 STAIRS module
3. SCS.TEST.PLILIB PLILIB to compile SCS modules, which contains:

SCWAPLI	CCL Transaction Work Area (PL/I)
TSDPLYRD	Temporary Storage Display Record Descriptor Dummy Section
TSDPLYR0	Temporary Storage Display Record 0 DSECT
TS047RD	DSECT for Temporary Storage Record written by DLN047 STAIRS module
4. SCS.TEST.SOURCE which contains:

CCLHELP	Help messages (input to CCL02 program)
CCL01	Correspondence table builder
CCL02	Creates CCL HELP file
CONNECT	Sign on program
DCT	Sample DCT
DLNCMDEF	STAIRS DLNCMDEF table input deck
DLNMSG	modified or added STAIRS messages (long)
DLNMSGSH	modified or added STAIRS messages (short)
DLN002	Updated module
DLN005	Updated module
DLN007	Updated module
DLN010	Updated module
DLN013	Updated module
DLN020	Updated module
DLN025	Updated module
DLN047	Updated module
EURODBC	DBC for EURO (dummy) database

FCT	Sample FCT
INISFORM	Sample input for correspondence table (CCL01)
PCT	Sample PCT
PPT	Sample PPT
SCH50501	Short SCS50501 map
SCH50601	Short SCS50601 map
SCH50700	Short SCS50700 map
SCH80001	Short SCS80001 map
SCH80101	Short SCS80101 map
SCH80102	Short SCS80102 map
SCH80103	Short SCS80103 map
SCH80400	Short SCS80400 map
SCH80401	Short SCS80401 map
SCH80402	Short SCS80402 map
SCSCMDEF	CCL Commands definition table input deck
SCS501	BASE Command Module
SCS502	FIND Command First Analyser
SCS503	SHOW Command Module
SCS504	PRINT Command Module
SCS505	HELP Command Module
SCS50501	SCS50501 map
SCS506	DISPLAY Command Module
SCS50601	SCS50601 map
SCS507	DEFINE Command Module
SCS50700	SCS50700 map
SCS508	SAVE Command Module
SCS509	DELETE Command Module
SCS510	INFO Command Module
SCS511	NEWS Command Module
SCS551	FIND Command: search history
SCS552	FIND Command: stack builder
SCS553	FIND Command: stack analyzer
SCS80001	Map
SCS80101	Map
SCS80102	Map
SCS80103	Map
SCS80400	Map
SCS80401	Map
SCS80402	Map
VIEMSG	CCL Messages
VIEMSGSH	CCL Short Messages
VIEPARMS	VIESGEN macro parameters
VIE740	Updated Module
VIE741	Updated Module
VIE801	Updated Module
VIE803	Updated Module
VIE804	Updated Module
VIE900	Updated Module

5. SCS.TEST.LOADLIB LOADLIB SCS with STAIRS 2.5 PTF D0004
 TLS 1.0 APAR corrected until end April 80
 + local FIX for SEARCH
 CICS 1.4 PTF 601 (pre-generated system)
 PL/I Optimizing Compiler, Version 1,
 Release 3.0, PTF 69
6. SCS.TEST.CARDS SOURCE and MACLIB update cards for SCS
- | | |
|----------|---|
| CCLTAB | INIS database Correspondence Table Creation
Sample JCL |
| DFHTCCLC | TTY Support (CICS Update) |
| DFHTCEXT | TTY Support (CICS Update) |
| DFHTCTRN | TTY Support (CICS Update) |
| DFHTCTWX | TTY Support (CICS Update) |
| DLNSTART | Update cards |
| DLN002 | Update Cards |
| DLN005 | Update Cards |
| DLN007 | Update Cards |
| DLN010 | Update Cards |
| DLN013 | Update Cards |
| DLN020 | Update Cards |
| DLN025 | Update Cards |
| DLN047 | Update Cards |
| GENMSGDL | STAIRS messages generation procedure |
| GENMSGSC | CCL messages generation procedure |
| LIST | Sample libraries list job |
| SCSC | CCL Modules compile procedures (input from
SCS.TEST.SOURCE) |
| SCSCOMP | CCL Modules update and compile procedure
(input from SCS.TEST.CARDS) |
| SCSMAC | MACLIB Update procedure |
| SCSMGEN | CCL Maps Generation Procedure (input from
SCS.TEST.SOURCE) |
| START | Sample CICS JOB |
| TWX020 | TTY Support (DLN020 update cards) |
| VIE740 | Update Cards |
| VIE741 | Update Cards |
| VIE801 | Update Cards |
| VIE803 | Update Cards |
| VIE804 | Update Cards |
| VIE900 | Update Cards |

7. SCS.HELP.MESSAGE CCL Help messages file (ISAM)
8. TNDXEURO Textindex file for EURO database
9. TEXTEURO Text file for EURO database
10. INVTEURO Inverted file for EURO database
11. DICTEURO Dictionary file for EURO database

Note that the distributed LOADLIB has the TRACE=YES option; this considerably impacts performance and is therefore not suitable in a production environment.

If TRACE=NO is desired, the DIN020, VIE800, and VIE900 modules must be reassembled.

Whenever problems occur in the execution of programs written in PLI (SCS5xx) using the distributed LOADLIB, these programs must be recompiled as described in point 7 of the installation instructions.

All the SCS.TEST.xxx files have been produced using the standard IEBCOPY IBM utility. The CCL.HELP.MESSAGE file has been unloaded using the IEBISAM utility. The other files have been produced using a standard IEBGENER IBM utility.

The suggested space allocations for the data sets are as follows:

Data Set	BLKSIZE	LRECL	RECFM	SPACE	DSORG
SCS.TEST.MACLIB	6400	80	FB	6400,(50,10,10)	PO
SCS.TEST.PLILIB	400	80	FB	6400,(10,10,10)	PO
SCS.TEST.SOURCE	6400	80	FB	6400,(200,50,10)	PO
SCS.TEST.LOADLIB	6420	6420	U	6420,(50,10,10)	PO
SCS.TEST.CARDS	1600	80	FB	1600,(90,20,10)	PO
SCS.HELP.MESSAGE	1612	1612	F	CYL,(1)	IS
K358.SCS.TNDX	1900	19	FB	CYL,(1)	DA
K358.SCS.TEXT	1954	1954	F	CYL,(1)	DA
K358.SCS.INVT	1952	1952	F	CYL,(1)	DA
K358.SCS.DICT	1952	1952	F	CYL,(1)	DA

To install the CCL implementation on STAIRS/VS the following steps must be taken:

1. Space must be allocated for the data sets;
2. The content of the distribution tape must be copied;
3. The new PPT, PCT, FCT and DCI must all be compiled. The entries required can be easily deduced from the examples given in SCS.TEST.SOURCE;
4. The new message files (SHORT and LONG) must be generated. The input cards are contained in SCS.TEST.SOURCE. An example of the procedure is given in the SCS.TEST.CARDS file. The file CCL.HELP.MESSAGE must be unloaded. The message input cards are contained in the CCLHELP member of SCS.TEST.SOURCE;
5. The DBCB and FFFILE for the EURO database must be generated and the relative entries must be added to the UREG records. The appropriate cards are given in the EURODBCB member of SCS.TEST.SOURCE.
6. In order to have TTY support, the DPHTCCLC, DPHTCEXT, DPHTCRN, and DPHTCTWX members in CICS.SOURCE must be updated using the corresponding members in the SCS.TEST.CARDS and the DFHTCF must be recompiled. The DLN020 in the STAIRS.SOURCE must also be updated with the cards contained in the TWX020 member of the SCS.TEST.CARDS;
7. If the CICS, STAIRS, TLS or PL/I levels differ from those indicated, the modified modules must be recompiled using a procedure which is analogous to that given in

the SCSCOMP member of the SCS.TEST.CARDS;

8. The start-up CICS deck must be modified by adding the SCS.TEST.LOADLIB as the first data set defined in the DFHRPL DD card; the VIEMSGSC, VIEMSGSH, CCLHELP and the EURO database DD cards must be added and the DDS required for print queues must be included.

A. SCSTWA

A specific SCS Transaction Work Area has been defined.

The layout of this SCSTWA is:

Assembler version

SCSFLAG	DS	XL1	GENERAL SCS FLAG	CCL1
SCSIND	EQU	X'80'	SCS ENVIRONMENT	CCL1
SCSCONV	EQU	X'40'	CONVERSATIONAL MODE	CCL1
SCSSYSHP	EQU	X'20'	SHORT PROMPTING SYSTEM	CCL1
SCSCMSHP	EQU	X'10'	SHORT PROMPTING COMMAND	CCL1
SCSPGISF	EQU	X'01'	INPUT SIM. FOR PAGE CMD	CCL1
*SCSPFIND	EQU	X'02'	FIND COMMAND GIVEN	CCL1
SCSCMRCV	EQU	X'04'	SCS COMMAND RECEIVED	CCL1
SCSNFND	EQU	X'08'	NO FIND INDICATOR	CCL1
SCSFLAG1	DS	XL1	GENERAL SCS FLAG	CCL1
SCSTSGN	EQU	X'10'	SCST SIGNED ON	CCL1
SCSBASE	DS	XL1	BASE FLAG	CCL1
SCSBSSN	EQU	X'01'	SIGN ON IN PROGRESS	CCL1
SCSBSNM	EQU	X'02'	DBNM SUPPLIED IN BASE COMMAND	CCL1
SCSBSNE	EQU	X'04'	DBNM ERROR IN BASE MODULE	CCL1
SCSBSPW	EQU	X'08'	DB PW SUPPL. IN BASE COMMAND	CCL1
SCSBSTH	EQU	X'10'	THESAURUS NAME GIVEN	CCL1
SCSBSTE	EQU	X'20'	THESAURUS NAME ERROR	CCL1
SCSBSTL	EQU	X'40'	THESAURUS LANGUAGE SUPPLIED	CCL1
SCSDSPY	DS	XL1	DISPLAY FLAG	CCL1
SCSDPYON	EQU	X'80'	DISPLAY ON	CCL1
SCSDPYFT	EQU	X'40'	DISPLAY FREE TEXT	CCL1
SCSDPYCT	EQU	X'20'	DISPLAY CONTROLLED TERM	CCL1
SCSQRYN	DS	H	SCS QUERY NUMBER	CCL1
SCSSAVNM	DS	CL4	NAME FOR SAVE COMMAND	CCL1
SCWACDN	DS	H	CURRENT DISPLAY NUMBER	CCL1
	ORG	SCWACDN		CCL1
SCSTHES	DS	CL4	THESAURUS NAME	CCL1
SCSTL	DS	CL1	THESAURUS LANGUAGE	CCL1
SCWAOFLG	DS	XL1	PRINT FLAG	CCL1
SCSPRIN	EQU	X'80'	PRINT COMMAND GIVEN	CCL1
SCSPRRN	EQU	X'40'	REMOTE PRINTING	CCL1

SCSPROF	EQU	X'20'	OFFLINE PRINTING	CCL1
SCSPRDS	EQU	X'10'	DISK PRINTING	CCL1
SCSSVBR	DS	F	SAVE AREA FOR BASE REG DLN020	CCL1
SCWARSVA	DS	F	SAVE AREA FOR REGISTER	CCL1
SCWADBNM	DS	CL4	DBNM (FILLED BY BASE MODULE)	CCL1
SCWADBPW	DS	CL8	DB PASSWORD (BASE MODULE)	CCL1
	ORG	SCWADBPW		CCL1
SCWATSK	DS	CL8	SCS TS KEY	CCL1
		SPACE 2		
-----				CCL1
*	DISPLAY COMMAND WORK AREAS (8 BYTES)			* CCL1
-----				CCL1
	ORG	SCWADBPW		CCL1
SCWADROA	DS	F	DISPLAY RECORD 0 AREA ADDRESS	CCL1
SCWADDRA	DS	F	DSPLY DESCRIPTOR RCD AREA ADDR.	CCL1
		SPACE 2		
SCWASTDN	DS	H	START DOCUMENT NUMBER (SHOW)	CCL1
SCWAENDN	DS	H	END DOCUMENT NUMBER (SHOW)	CCL1
SCWAINDN	DS	H	INCREMENT DOCUMENT NUMBER (SHOW)	CCL1
SCSDELIN	DS	C	CHAR FOR DEL LINE	CCL1
SCSBACSP	DS	C	CHAR FOR BACK SPACE	CCL1
SCWABTAP	DS	F	EASE AREA AVAIL. POS.	CCL1
SCWABTNM	DS	CL160	BASE NAME AREA	CCL1
	ORG	SCWABTNM		CCL1
SCWATSRC	DS	CL160	I.S. QUERY RECORD	CCL1
CCLATFLD	DS	F	ADDRESS OF P.C. TABLE	
CCLATTH	DS	F	ADDRESS OF THES. RELATOR TABLE	
CCLASTK1	DS	F	ADDRESS OF FIRST STACK ELEMENT	
CCLASTK2	DS	F	ADDRESS LAST STACK ELEMENT	
CCLASTK3	DS	F	ADDRESS OF CURRENT STACK ELEMENT	
CCLADDR1	DS	F	ADDRESS OF WORK AREA	
CCLADDR2	DS	F	ADDRESS OF WORK AREA	
CCLADDR3	DS	F	ADDRESS OF WORK AREA	
CCLPCLN	DS	H	REAL LENGTH OF PARAGRAPH NAME	
CCLPLEV	DS	H	PARENTHESIS LEVEL	
CCLFP1	DS	H	INTERNAL PARAMETER	
CCLFP2	DS	H	INTERNAL PARAMETER	
CCLFP3	DS	H	INTERNAL PARAMETER	
CCLFP4	DS	H	INTERNAL PARAMETER	
CCLFP5	DS	H	INTERNAL PARAMETER	
CCLLP1	DS	H	LENGTH OF AREA BASED ON CCLADDR1	
CCLLP2	DS	H	LENGTH OF AREA BASED ON CCLADDR2	
CCLLP3	DS	H	LENGTH OF AREA BASED ON CCLADDR3	
CCLFRC	DS	H	RETURN CODE FROM INTERNAL ROUTINE	
CCLTSRCH	DS	CL1	SEARCH FUNCTION	
ROUTCD	DS	CL1	INTERNAL ROUTINE NAME	
CCLFFLAG	DS	XL1	SEARCH FLAG	
CCLINQRY	EQU	X'01'	SCS INTERNAL QUERY GIVEN	
CCLFFIND	EQU	X'02'	FIRST FIND EXECUTION	
QNCHAR	DS	CL5	QUERY NUMBER IN CHAR	CCL1

	ORG	SCWATSRC		
SCWATSLN	DS	H	T. S. RECORD LENGTH	CCL1
SCWANUSD	DS	CL2	ZERO BINARY AREA	CCL1
SCWATSID	DS	0CL8	SCS T. S. IDENTIFICATION	CCL1
SCWAUSID	DS	CL2	INTERNAL USER ID	CCL1
SCWARCID	DS	CL2	TYPE OF RECORD	CCL1
SCWALNNO	DS	H	SCS QUERY NUMBER	CCL1
SCWASQNC	DS	H	LINE QUERY NUMBER	CCL1
SCWATSTL	DS	H	USER QUERY NUMBER STAIRS	CCL1
SCWAQRY	DS	CL146	QUERY'S SAVE AREA	CCL1
	ORG	SCWAQRY		CCL1
SCWACOMM	DS	CL8	SCS COMMAND	CCL1
SCWASQRY	DS	CL138	SCS COMMAND'S PARAMETERS	CCL1
	ORG	SCWABTAP		CCL1
SCWAPRAP	DS	F	ADDRESS OF PRINT COMMAND	CCL1
	ORG	SCWARSVA		CCL1
SCWAHALF	DS	H	HALFWORD WORK AREA	CCL1
SCWAHLF1	DS	H	2. ND HALFWORD WORK AREA	CCL1

PL/I version

/* 2 VIETWA		VIETWA FILLER TO BE INSERTED IN CODE	
2 SCSTWA,		/* SCSTWA	CCL1
3 SCSFLAG BIT (8),		/* GENERAL SCS FLAG	CCL1
3 SCSFLAG1 BIT (8),		/* GENERAL SCS FLAG	CCL1
3 SCSBASE BIT (8),		/* BASE FLAG	CCL1
3 SCSDSPLY BIT (8),		/* DISPLAY FLAG	CCL1
3 SCSQRYN BIN FIXED (15),		/* SCS QUERY NUMBER	CCL1
3 SCSSAVNM CHAR (4),		/* NAME FOR SAVE COMMAND	CCL1
3 SCSTHES CHAR (4),		/* THESAURUS NAME	CCL1
3 SCSTL CHAR (1),		/* THESAURUS LANGUAGE	CCL1
3 SCWAOFLG BIT (8),		/* PRINT FLAG	CCL1
3 SCSSVBR BIN FIXED (31),		/*SAVE AREA FOR BASE REG DLN020	CCL1
3 SCWARSVA BIN FIXED (31),		/* SAVE AREA FOR REGISTER	CCL1
3 SCWADB NM CHAR (4),		/* DBNM (FILLED BY BASE MODULE)	CCL1
3 SCWADBPW CHAR (8),		/* DB PASSWORD (BASE MODULE)	CCL1
3 SCWASTDN BIN FIXED (15),		/* START DOCUMENT NUMBER (SHOW)	CCL1
3 SCWAENDN BIN FIXED (15),		/* END DOCUMENT NUMBER (SHOW)	CCL1
3 SCWAINDN BIN FIXED (15),		/* INCREMENT DOC NUMBER (SHOW)	CCL1
3 SCSDELIN CHAR (1),		/* CHAR FOR DEL LINE	CCL1
3 SCSBACSP CHAR (1),		/* CHAR FOR BACK SPACE	CCL1
3 SCWABTAP BIN FIXED (31),		/* BASE AREA AVAIL. POS.	CCL1
3 SCWABTNM CHAR (160),		/* BASE NAME AREA	CCL1
3 CCLATFLD POINTER,		/* ADDRESS OF P.C. TABLE	*/

```

3 CCLATTH POINTER, /* ADDRESS OF THES. RELATOR TABLE */
3 CCLASTK1 POINTER, /* ADDRESS OF FIRST STACK ELEMENT */
3 CCLASTK2 POINTER, /* ADDRESS LAST STACK ELEMENT */
3 CCLASTK3 POINTER, /* ADDRESS OF CURRENT STACK ELEMENT */
3 CCLADDR1 POINTER, /* ADDRESS OF WORK AREA */
3 CCLADDR2 POINTER, /* ADDRESS OF WORK AREA */
3 CCLADDR3 POINTER, /* ADDRESS OF WORK AREA */
3 CCLPCLN BIN FIXED(15), /* REAL LENGTH OF PARAGRAPH NAME */
3 CCLPLEV BIN FIXED(15), /* PARENTHESIS LEVEL */
3 CCLFP1 BIN FIXED(15), /* INTERNAL PARAMETER */
3 CCLFP2 BIN FIXED(15), /* INTERNAL PARAMETER */
3 CCLFP3 BIN FIXED(15), /* INTERNAL PARAMETER */
3 CCLFP4 BIN FIXED(15), /* INTERNAL PARAMETER */
3 CCLFP5 BIN FIXED(15), /* INTERNAL PARAMETER */
3 CCLLF1 BIN FIXED(15), /* LENGTH OF AREA BASED ON CCLADDR1 */
3 CCLLF2 BIN FIXED(15), /* LENGTH OF AREA BASED ON CCLADDR2 */
3 CCLLF3 BIN FIXED(15), /* LENGTH OF AREA BASED ON CCLADDR3 */
3 CCLFRC BIN FIXED(15), /* RETURN CODE FROM INTERNAL ROUTINE */
3 CCLTSRCH CHAR(1), /* SEARCH FUNCTION */
3 ROUTINE_CODE CHAR(1), /* INTERNAL ROUTINE NAME */
3 CCLFFLAG BIT(8) UNALIGNED, /* SEARCH FLAG */
3 QNCHAR PIC'ZZZZ9'; /* QUERY NUMBER IN CHAR CCL1 */
DCL 1 SCWATSRC /* T.S. QUERY RECORD CCL1
  BASED(SCWAPTR1), /* REDEFINE BASE NAME AREA CCL1
2 SCWATSLN CHAR(2), /* LENGHT OF RECORD CCL1
2 SCWANUSD CHAR(2), /* ZERO BINARY CCL1
2 SCWATSID, /* T.S. IDENTIFICATION CCL1
3 SCWAUSID CHAR(2), /* INTERNAL USER IDENTIFICATION CCL1
3 SCWARCID CHAR(2), /* TYPE OF RECORD CCL1
3 SCWALNNO CHAR(2), /* SCS QUERY NUMBER CCL1
3 SCWASQNO CHAR(2), /* LINE QUERY NUMBER CCL1
2 SCWATSTL BIN FIXED(15), /* USER QUERY NUMBER STAIRS CCL1
2 SCWAQRY, /* SCS QUERY SAVE AREA CCL1
3 SCWACOMM CHAR(8), /* SCS COMMAND CCL1
3 SCWASQRY CHAR(138); /* SCS COMMAND'S PARAMETERS CCL1
DCL 1 SCWAWORK /* HALFWORDS WORK AREA CCL1
  BASED(SCWAPTR3), /* REDEFINE SAVE AREA REG. CCL1
  2 SCWAHALF BIN FIXED(15), /* HALFWORD WORK AREA CCL1
  2 SCWAHLF1 BIN FIXED(15); /* 2. ND HALFWORD WORK AREA CCL1
DCL SCWASHW CHAR(160) /* SHOW QUERY AREA CCL1
  BASED(SCWAPTR1) /* REDEFINE BASE NAME AREA CCL1
DCL SCWAPRAP BIN FIXED(31) /* ADDRESS OF PRINT COMMAND CCL1
  BASED(SCWAPTR2) /* REDEFINE BASE AV. POSITION CCL1
DCL SCWATSK CHAR(8) /* REDEFINE DB PW AS SCS TS KEY CCL1
  BASED(SCWAPTR4); /*
DCL 1 SCWADPLY_WORK /* DISPLAY VARIABLES CCL1 */
  BASED(SCWAPTR4), /*
  2 SCWADROA POINTER, /* DISPLAY REC 0 POINTER CCL1 */
  2 SCWADDRA POINTER; /* DISPLAY DESC. REC. POINTER CCL1 */
DCL SCWACDN BIN FIXED(15) /* CURRENT DISPLAY NUMBER CCL1 */

```

```

        BASED (SCWAPTR5);
SCWAPTR1=ADDR(SCWABTNM);      /* SET POINTER ADDRESS      CCL1
SCWAPTR2=ADDR(SCWABTAP);      /* SET POINTER ADDRESS      CCL1
SCWAPTR3=ADDR(SCWARSVA);      /* SET POINTER ADDRESS      CCL1
SCWAPTR4=ADDR(SCWADBPW);      /* SET POINTER ADDRESS      CCL1
SCWAPTR5=ADDR(SCSTHES);      /* SET POINTER ADDRESS      CCL1 */
%PAGE;
/*****
/*  FLAGS IN SCSFLAG
/*****
DECLARE
SCSIND  BIT(8)  INIT('10000000')  STATIC,/*  SCS ENVIRONMENT  CCL1 */
SCSCONV BIT(8)  INIT('01000000')  STATIC,/*  CONVERS.  MODE  CCL1 */
SCSSYSHP BIT(8) INIT('00100000')  STATIC,/*  SH PROMPT SYS  CCL1 */
SCSCMSHP BIT(8) INIT('00010000')  STATIC,/*  SH PROMPT CMD  CCL1 */
SCSPGISF BIT(8) INIT('00001000')  STATIC,/*  INP SIM.PAGE CMD CCL1 */
/*SCSFFIND BIT(8) INIT('00000100')  STATIC,/*  FIND CMD GIVEN  CCL1 */
SCSCMRCV BIT(8) INIT('00000010')  STATIC,/*  SCS CMD RECEIVED CCL1 */
SCSNFND  BIT(8) INIT('00000001')  STATIC; /*  NOFIND INDICATOR CCL1 */
/*****
/*  FLAGS IN SCSFLAG1
/*****
DECLARE
SCSTSGN BIT(8) INIT('00010000')  STATIC; /*  SCST SIGNED ON  CCL1 */
/*****
/*  FLAGS IN SCSBASE
/*****
DECLARE
SCSBSSN BIT(8) INIT('00000001')  STATIC,/*  SIGNON IN PROGRES CCL1 */
SCSBSNM BIT(8) INIT('00000010')  STATIC,/*  DBNM SUPPLIED  CCL1 */
SCSBSNE BIT(8) INIT('00000100')  STATIC,/*  DBNM ERROR  CCL1 */
SCSBSPW BIT(8) INIT('00001000')  STATIC,/*  DB PW SUPPL.D  CCL1 */
SCSBSTH BIT(8) INIT('00010000')  STATIC,/*  TH NAME GIVEN  CCL1 */
SCSBSTE BIT(8) INIT('00100000')  STATIC,/*  TH NAME ERROR  CCL1 */
SCSBSTL BIT(8) INIT('01000000')  STATIC; /*  TH LANG SUPPL.D CCL1 */
/*****
/*  FLAGS IN SCSDSPLY
/*****
DECLARE
SCSDPYON BIT(8) INIT('10000000')  STATIC,/*  DISPLAY ON  CCL1 */
SCSDPYFT BIT(8) INIT('01000000')  STATIC,/*  DISPLAY FREE TEXT CCL1 */
SCSDPYCT BIT(8) INIT('00100000')  STATIC; /*  DSPLY CONTR. TERM CCL1 */
/*****
/*  FLAGS IN SCSOFLG (OUTPUT FLAG)
/*****
DECLARE
SCSPRIN BIT(8) INIT('10000000')  STATIC,/*  PRINT CMD GIVEN  CCL1 */
SCSPRNR BIT(8) INIT('01000000')  STATIC,/*  REMOTE PRINTING  CCL1 */
SCSPROP BIT(8) INIT('00100000')  STATIC,/*  OFFLINE PRINTING CCL1 */
SCSPRDS BIT(8) INIT('00010000')  STATIC; /*  DISK PRINTING  CCL1 */

```

```

/*****
/*   FLAGS IN CCLFFLG (FIND FLAG)
/*****
DECLARE
CCLINQRY BIT(8) INIT('00000001') STATIC,/*SCS INT.  QRY GIVEN CCL1*/
CCLFFIND BIT(8) INIT('00000010') STATIC;/* FIRST FIND EXECUTION CCL1*/

```

B. Correspondence Table

A table called CCL<dbname> has been built to denote the correspondences between the CCL field labels and the STAIRS paragraph names and/or formatted field names.

The format of this table is as follows:

First entry (only one type).

Type	Privacy level	Field label		Low p.c.	Up p.c.	Paragraph code
(=F)	(=255)	blank	'FCBFIELD'			
<- 1 ->	<- 1 ->	<- 2 ->	<-- 8 -->	< 2 >	< 2 >	<-- 4 -->

bytes

Entry (first type=formatted-field).

Type	Privacy level	Field label	STAIRS name (FFIELD)	LL	Pos	FT	Par. code
(=F)							
<- 1 ->	<- 1 ->	<- 2 ->	<-- 8 -->	< 2 >	< 2 >	< 1 >	<- 3 ->

bytes

Entry (second type=paragraph).

Type	Privacy level	Field label	STAIRS name (paragraph name)
(=P)			
<- 1 ->	<- 1 ->	<- 2 ->	<-- 16 -->

bytes

Last entry.

'* STOPPER TABLE'
<----- 20 ----->

bytes

In order to build this table, the CCL01 program must be

executed according to the example given in the CCLTAB member of SCS.TEST.CARDS.

The format of the input cards is as follows:

First card:

col. 1-3 lower limit

col. 4-6 upper limit

Note that these two limits define the special paragraph class name 'FORMFIELD'. This range must be at least equal to the (number of formatted fields + 1) and must not overlay other paragraph classes.

Successive cards:

col. 1-2 CCL field label

col. 3-3 blank

col. 4-19 name of corresponding STAIRS paragraph or FFIELD

The data base name is given as a parameter.

C. Implementation Logic

As a general rule, the SCS commands have been implemented as follows:

- the input string which contains the SCS command is passed to an ad hoc PL/I module. The string is analysed and all its parameters are scanned. In general, the input is not positional. If the same parameter is entered more than once, the most recent input is assumed as that valid. Any errors are indicated.

All valid parameters are put in SCSTWA and a flag is switched on.

The input string is modified into an acceptable STAIRS format and control is returned to the STAIRS command driver.

The appropriate STAIRS modules are modified so that their logical flow is regulated by flags (e.g. simulating inputs and masking outputs).

D. General support modifications

The general support modifications are all those relative to CICS and to the two MACRO service modules DLN020 and VIE900, which concern:

- TTY support
- Input conversion
- Conversational / nonconversational input
- SCS command definitions
- SHORT/LONG dialog (prompting)
- Message file selection
- Map selection.

D1. TTY Support

The TTY support has been implemented as follows:

- Time out (EP)

The BTAM issues a READ CONVERSATIONAL (i.e. a READ with time-out). This problem can be bypassed by defining the parameter TEXTTO in the macro GROUP of EP as equal to NONE or 0 depending on whether the 3705 is working in PEP or EP mode.

- Input handling (DLN020, EP, TCTWX)

To send a message from the TWX terminal to the computer the following steps are necessary:

1. Press the "CONTROL" and "ALPHA" keys for EOM
2. Press the "RETURN" key
3. Press the "LF" key.

These three functions can also be achieved by just pressing the "RETURN" key.

The DLN020 module has been modified so that a CR-LF can be sent after a READ.

In EP the CHAREC=(XONOFF,B1) parameter has been defined.

The DFHTCTWX module has been modified so that "RETURN" can be accepted as EOM.

- Translate table (DFHTCTRN)

The standard CICS translate table for TWX terminals

does not include all the possible upper and lower case characters or the transliteration for even and odd parity.

A new translate table is available.

- NL support (DFHTCEXT)

The sequence of characters X'1517' has been converted to X'1526'.

- System prompting

With CCL the system is ready for input when the '/?' sequence is sent.

This implementation has been achieved by modifying the string sent by the TCP at READ CONVERSATIONAL time.

The standard string has been changed to '/?'.

This change has necessitated an update to the DFHTCCLC member of the CICS.SOURCE and a regeneration of the DFHTCP module.

D2 = Input conversion (DLN020)

The CCL dialog is essentially non-conversational, i.e. input normally consists of a command followed by parameters. STAIRS interprets all input beginning with ".." as commands. Consequently, each input is modified by prefacing it with two dots. The input is processed by STAIRS as a command, and can then be passed to a specific module (see DLNCMDEF

STAIRS macro).

D3 - Conversational / nonconversational input (DLN020)

When the system asks for a specific parameter (e.g. the Database name, the password, the thesaurus name) the input given in reply must not be interpreted as a command.

In this case, the input conversion described above would be mistaken and lead to an endless loop. Therefore, the SCSCONV flag has been introduced. If this flag is switched on, no input conversion takes place. However, as CCL requires that the STOP command can be entered at any time, this word is checked in both conversational and non-conversational mode. In both cases, the input is converted to ..OFF.

This is in agreement with the STAIRS philosophy, which in certain cases accepts the input ..OFF in order to exit from an endless question-answer loop.

D4 - CCL command definitions (DLN020, DLN010)

All the SCS commands are defined in the SCSCMDEF table, which is assembled using the DLNCMDEF macro.

The SCSCMRCV switch is turned on for each input in the CCL environment, and the command driver skips the normal input control to check whether a STAIRS command has been issued.

It compares the input with the SCSCMDEF table. Incorrect and non-existent commands are refused. If the command is found in the table, control is passed to the relevant module and the SCSCMRCV switch is turned off. In this way, the command driver can examine the STAIRS command into which the ad hoc module has transformed the input and can process it in the standard STAIRS mode.

D5 - SHORT/LONG Dialog (DLN020)

The SHORT/LONG prompting is governed by two switches: SCSSYSHP (SYstem SHort Prompting) and SCSCMSHP (CoMmand SHort Prompting) as the short prompting can either be imposed with the DEFINE command (and in this case will remain valid for the whole session) or by preceding the command by a dot "." (valid for the execution of the command). Management at command level is realised by the DLN020 module which switches the SCSCMSHP off for each input, and then on again if the input string should begin with a dot.

D6 - Message File Selection (DLN020)

STAIRS and TLS messages, not defined directly in the modules, are contained in two files which have DLNMSG and

VIEMSG, respectively, as their DDNAME.

The CCL messages are contained in 4 files: DLNMSGSC, DLNMSGSH, VIEMSGSC, VIEMSGSH. Whenever STAIRS or TLS use the DLN020 module to read a message from the DLNMSG or VIEMSG files in the SCS environment, the file name is changed to DLNMSGSH or VIEMSGSC. If at least one of the SCSSYSHP or SCSCMSHP flags is switched on, either the DLNMSGSH or the VIEMSGSH file is read, otherwise either the DLNMSGSC or VIEMSGSC file is read.

If the message is not found in either of the two files, the standard files are read.

In this way, the four SCS files which contain all the modified messages can be defined without duplicating the original files.

D7 - Map Selection (VIE900)

The maps are invoked by the modules with standard names, i.e. VIE $nnnn$, where nnn is the module number and nn is the number of the map. In the SCS environment, the map name is changed to SCS $nnnn$. If at least one of the SCSSYSHP or SCSCMSHP flags is on, the name of the map is changed to SCH $nnnn$. The messages recalled by the maps can be found in the appropriate files owing to the modifications to the DLN020 module (see message file selection).

E - Command Implementation

All the commands which have been implemented are described in this chapter. For each command is given:

- a description of the acceptable parameters and their significance;
- a brief description of the implementation.

Certain factors which are common to the entire implementation are listed here below:

- the implementation supports terminals of the 3270 type in addition to TTY type terminals;
- on TTY type terminals, the system advises when it is ready to receive input by sending '/?';
- except when the system makes specific queries, each input is interpreted as a command. STOP is always interpreted as a command;
- all the commands can be preceded by a stop (.), which indicates that in the execution of that command the SHORT dialog is preferred (experienced user).

E1 - Sign on (CONNECT, DLN007, VIE801, VIE804)

The sign-on procedure has been implemented by coding an ad hoc CICS transaction called CONN.

The input format is as follows:

```
|-----|
|          CCL
| CONN[ECT] CNUCE  TLS      [ user-password user-name ]
|          STAIRS
|-----|
```

Abbreviated forms of CONNECT (CONN, CONNE, CONNEC) are accepted as valid input. Node indication is mandatory (i.e. CNUCE).

If erroneous parameters are entered, the user is given a list of the available transactions:

SCST for information retrieval using CCL

AQTL for information retrieval using STAIRS/VS - TLS

AQUA for information retrieval using STAIRS/VS

and is requested to enter the code for the transaction he requires.

Valid parameters are CCL, TLS or STAIRS. For CCL or TLS these parameters are followed optionally by the user password and name. If STAIRS is entered, name and password are mandatory.

When one of these parameters is entered, the appropriate transaction code (optionally followed by user password and name) is placed in the TIOA from position TIOADBA and control is passed to the DLN007 module by means of an XCTL; the normal STAIRS or TLS sign-on procedure then takes place.

If no parameter is entered, CCL is assumed by default.

In the DLN007 module, when the active transactions are CONN or SCST, the SCSIND (SCS environment active) and the SCSCONV (conversational input) bits are switched on in the SCSFLAG flag, and the SCSBSSN bit is switched on in the SCSBASE (sign on in progress) flag.

The conversational input bit must be switched on before the user password and name can be accepted if they have not already been entered together with the transaction code. This bit is switched off before exiting from the DLN007 module.

The SCSBSSN bit in the SCSBASE flag regulates the logical flow in the VIE801 module, therefore, the user is automatically accessed to the EURO data base (which is not associated to a thesaurus) but remains completely unaware of this.

At this point, sign on procedure is complete and the user can either issue a command, or press the "enter" key, thus passing the control to the VIE804 module and

obtaining a map which displays the permitted commands.

E2 - HELP Command (SCS505)

The HELP command gives the user general advice on the use of the system.

H and ? are also accepted as valid inputs.

The command format is:

Command	Parameter	Default value
H[ELP] or ?	[command]	HELP

Where:

command = any valid CCL command

If the command is issued without parameters, HELP or ? is assumed, and the system displays a list of allowed inputs.

If incorrect input is given:

HELP HELP

is assumed.

The user can move backward or forward using the BACK, MORE and PAGE commands.

SCS505 reads messages from CCLHELP file, and displays them.

E3 - BACK Command (DLN020)

During a DISPLAY, SHOW or HELP, the BACK command allows the user to move backward to review a preceding page.

B is also accepted as valid input.

The command format is:

Command	Parameter	Default value
B[ACK]	[number]	1

Where:

number indicates how many pages the user want to move backward

The input is converted into:

p-n for 3270

and

doc-n for TTY

E4 - MORE Command (DLN020)

During a DISPLAY, SHOW or HELP, the MORE command allows the user to move forward to retrieve a successive page.

M is also accepted as valid input.

The command format is:

Command	Parameter	Default value
M[CR]	[number]	1

Where:

number indicates how many pages the user want to move forward

A "carriage return" is understood as:

MORE 1

The input is converted into:

p-n for 3270

and

doc-n for TTY

E5 = PAGE Command (DLN020)

During a DISPLAY, SHOW or HELP, the PAGE command allows the user to retrieve a specific page.

PA is also accepted as valid input.

The command format is:

Command	Parameter	Default value
PA[GE]	[number]	current page + 1

Where:

number indicates the page the user want to retrieve.

The input is converted into:

P=n

E6 - DEFINE Command (SCS507)

The DEFINE command is used to assign values to specific parameters of the system.

The abbreviated form DE is also valid input.

The command format is as follows:

Command	Parameter	Default value
DE[FINE]	[DL = char]	X'00'
	[;BS = char]	X'00'
	[;M = S(hort)/L(ong)]	LONG
	[;PAGE = (pl,ln,mg)]	24,24,0
	[;DEFAULT]	

Where:

DL defines the character used as "delete line" for TTY's
 BS defines the character used as "backspace" for TTY's
 M defines whether LONG or SHORT messages are desired
 PAGE defines the "page size" as:

pl = page length

ln = number of lines per page

mg = margin at the top of the page

The parameters are positional, the absence of a parameter is denoted by a comma. Missing parameters are calculated from the given values. The parentheses are mandatory.

If only one parameter is given, the parentheses are not mandatory and the values are taken as pl, setting ln=pl and mg=0.

DEFAULT resets all parameters to their default values.

The order of the parameters is unimportant. Should a parameter be specified more than once, the last specification is taken as that valid.

The DEFINE command is handled by the SCS507 module. The appropriate values are put in SCSTWA or DLNTWA.

E7 - STOP Command (DLN020)

The STOP command is used to log out of the system.

The command format is:

```
|-----|  
|  STOP  |  
|-----|
```

This command is directly handled by the DLN020 module. Whenever the STOP command is given, the input string is converted into ..OFF NOCONT and the user logs out of the system.

E8 - OWN Command (DLN002, DLN010)

The OWN command allows STAIRS/VS-TLS own commands to be used.

The command format is:

```
|-----|  
|  OWN  |  
|-----|
```

In order to return to the CCL environment, the user should enter the command ..CCL.

The OWN command is recognised by both the DLN002 and the DLN010 modules. In both cases, the SCSIND and SCSCMRCV bits in the SCSFLAG are turned off and a normal STAIRS/TLS session can take place.

E9 - BASE Command (SCS501, VIE801)

BASE selects the database that is to be searched.

BAS is also accepted as valid input.

The command format is:

```

|-----|
| BAS[ E]      [ dbname ]
|               [ ;P      = password ]
|               [ ;TL     = thesaurus language ]
|               [ ;THES  = thesaurus name or LIST ]
|-----|
| or
|-----|
| BAS[ E]      ?
|-----|

```

Where:

dbname name of database to be accessed (four characters)
P = indicates the database password, if any (max
8 characters)
TL= asks for a thesaurus in a specific language
(by default E = English)
THES= specifies the thesaurus name (four characters)
or
THES=LIST asks for a list of available thesauri
BASE ? asks for the name of the currently active database
and its structure (i.e. field labels)

Note that no parameter can be entered if the data base name is missing. If no parameter is entered, a list of available databases is shown, and the user is asked to select one of them (by name or number).

If THES=<thname> is missing and a thesaurus is associated to the database, this thesaurus is automatically selected.

If T=NONE is specified in the DBCB, no thesaurus is

selected.

The BASE command is processed by the SCS501 module. This module examines the parameters given with the command, moves them to appropriate fields of the SCSTWA and turns the necessary flags on. The input string is converted into ..CHANGE and the module then returns to DLN010, which calls VIE801.

VIE801 has been modified in order to mask prompting for parameters which have already been given in the BASE command and also to maintain a conversational input.

If essential parameters should be missing (e.g. the database name, the password or the thesaurus identification) a normal TLS dialog occurs.

The maps and the read routines have, however, been modified; the databases and the thesauri available are numbered so that they can be chosen either by their name or their order number in the output map.

When the user enters the command BASE ?, the name of the database in which the user is operating is displayed, the command driver returns to the VIE804 module and a list of all the possible commands is displayed.

E10 - DISPLAY Command (SCS506, VIE740, VIE741, DLN013, VIE803)

The DISPLAY command is used to obtain listings of logically related thesaurus terms or dictionary words in alphabetical order.

The abbreviated form D is also accepted as valid input.

In the FIND command references to the displayed terms can be made via the "T=" parameter.

The command format is:

```

-----|-----
| D[ISPLAY] | [CT [=] [thrl] descriptor ] |
|           | or                               |
|           | [FT] [=] term                    |
|           | or                               |
|           | <number>                          |
|           | or                               |
|           | ?                                  |
|-----|-----

```

Where:

CT asks for a display of logically related terms;
 FT asks for a display of dictionary words in alphabetical order;
 thrl = thesaurus relator, i.e. relator of a semantic field in the selected thesaurus to be displayed (as default ALL is assumed);
 descriptor = main descriptor of the semantic field to be displayed (may be masked with \$ sign);
 term = masked search term in the dictionary
 <number> = a number indicating any display already made.
 If later on during the session a reference with T= is made in a FIND command, it will be relative to

the display which is indicated by <number>
? asks for a display of the last successful display. If later on during the session a reference with T= is made in a FIND command, it will be relative to the last successful display.

The DISPLAY command is processed by the SCS506 module, which operates in two different ways, depending on whether a display of logically related terms or of alphabetically ordered words in the dictionary is requested.

With a logically related terms display (CT display), a LINK to VIE701 is executed.

The VIE740 module has been modified to take the input directly from an appropriate SCSTWA field, not from the terminal.

The VIE741 module has been modified so that a temporary storage record is written for each descriptor found. In addition, the map writing is bypassed, as a "carriage return" is simulated as input. On RETURN, a temporary storage "record 0" is written. If no descriptor is found, VIE740 simulates a "carriage return" and returns to VIE701 and to SCS506.

When there is a display of alphabetically ordered words in the dictionary display (FT display), the input is converted into:

```
ROOT <term> [TO <term2>]
```

and DLN013 is linked; this in turn links VIE803, etc.

If the DISPLAY function is active when DLN013 is ready to write, the module returns to SCS506.

The SCS506 module reads the temporary storage records written by DLN047, releases them, and writes in temporary storage a "record 0" plus one record for each term found.

At this point, SCS506 displays a map with sequentially numbered terms, containing relators and descriptors if there has been a CT display, or number of occurrences and words if there has been a FT display.

The user can move backward or forward using the paging commands BACK, MORE e PAGE.

E11 - FIND Command (SCS502, SCS551, SCS552, SCS553, VIE803, DLN013, DLN025)

The FIND command is used to enter search terms and search statements.

The abbreviated form F is also valid input.

The command format is:

```

|-----|-----|
| F[IND] | identifier [ operator identifier ] ... |
|-----|-----|
|         | OR |
|-----|-----|
| F[IND] | ? |
|-----|-----|

```

Where:

identifier =

- a) - a search term or code which may be truncated or restricted to a particular type (see later identifier modification);
- b) - a literal search phrase, enclosed in double quotes (e.g. "black and white");
- c) - a label identifying one or more search terms displayed at the terminal (e.g. T= n [TO m]). If the operator TO is used, the terms are logically OR'ed;
- d) - a label identifying one or more previous search statements (e.g. S= n [TO m]). Once again, if the operator TO is used, the terms are logically OR'ed.

and

operator=

- a) - any boolean logical connector (AND, OR, NOT).

Examples:

```

FIND smith AND wesson
FIND S=1 OR bond

```



```
    FIND T=3 TO 5 AND S=2 TO 4 NOT butterfly
```

If brackets are used to ensure that a sequence of operators is executed in the intended sequence, the logic within the brackets is executed first.

When the same logical operator is to be used to connect a number of terms, a shortened form of list notation can be used.

Example:

```
    FIND linus AND lucy AND schroeder AND snoopy  
is equivalent to:  
    FIND (AND linus;lucy;schroeder;snoopy)
```

and

FIND ? asks for a list of the previous searches

E3.1 - Identifier modification.

An identifier may be modified using a prefix or suffix to restrict searches to individual fields or to indicate special types of search terms.

E3.1.1 - Prefix.

- A prefix is divided into a field (or data element) label and a relational connector.
- The accepted relational connectors are:
= , < , > , <= , >=

For field labels which do not have linear ranges

(e.g. free text) only the connector = is valid.

- With field labels with tree structured ranges, in particular for structured thesauri, standard connectors are:

DOWN UP NT BT

Other connectors may be defined by the user in the table VIET<thname>

E3.1.2 - Suffix

- The search statement can be further qualified using the suffix facility.

Example:

FIND <fl1> = computer/<fl2>,<fl3>

where <fl1>, <fl2>, <fl3> are field labels.

Note that search statements referring to field labels with linear ranges may not be qualified using the suffix facility. Therefore, a search statement may be qualified using prefixes and suffixes only if the relational connector '=' or a thesaurus relator are used.

E3.1.3 - Prefix and parentheses

To avoid repeated use of a given prefix, a shortened form may be used.

Examples:

```
FIND <fl1> = butterfly AND <fl1> = daisy
```

is equivalent to:

```
FIND <fl1> = (butterfly AND daisy)
```

or:

```
FIND (butterfly AND daisy)/<fl1>
```

and:

```
FIND AU = (schultz OR parker OR hart)
```

is equivalent to:

```
FIND AU = (OR schultz;parker;hart)
```

or:

```
FIND (OR schultz;parker;hart)/AU
```

E3.2 - Adjacency connector

When searching free text, it is possible to specify that two words should occur in the same paragraph. This is achieved by entering:

<word1> ... <word2>

E3.3 - Truncation

The character masking symbol (\$) is used to search masked terms.

Example:

FIND compu\$

will retrieve computational, computer, computing, etc.

E3.4 - Syntax limitations

- Field labels with a linear range cannot be mixed with others in one query.
- Backreference in queries with field labels having linear ranges is allowed. However, the backreferenced querie(s) must appear at the beginning. A logical AND between backreferenced queries and entered identifiers is assumed, irrespectively of the actually entered operator.

e.g. FIND S=1 TO 3 OR na=10 AND py>=79

is converted into:

```
FIND S=1 TO 3 AND na=10 AND py>=79
```

The FIND command is examined by the SCS502, SCS551, SCS552 and SCS553 modules.

The VIE803 module has been changed in order to mask the READ of the query which is taken from a user acquired area.

The DLN013 module has been changed so that the CCL query, its number and the number of the last STAIRS query can be written in temporary storage.

The query number appears at the top of the printout of the results.

E12 - SHOW Command (SCS503, DLN005)

SHOW causes the retrieved documents to be displayed at the terminal.

S is also assumed as valid input. The input is not positional. If a parameter should be entered more than once, the last value entered is held to be valid.

The command format is as follows:

Command	Parameter	Default value
S[HOW]	[S = qn]	last query
	[;R = n [TO m]]	1 TO 5
	[;I = k]	1
	[;F = p1;p2;...]	
	or	ALL
	[;Fn]	

Where:

qn = query number
n = first document number in the list to be displayed
m = last document number in the list to be displayed
k = increment document number (for skinning list)
pn = field label (= STAIRS paragraph or formatted field)
Fn = predefined format

where "n" is a number ranging from 1 to 23.
These numbers are in correspondence with the formats "D" to "Z" that can be defined using the DLNPCDEF macro of STAIRS/VS-TLS.

The SHOW command is processed by the SCS503 module.

The search statement number, record number and record increment number are all stored in SCSTWA, while the

format is moved into a command area.

The table giving the correspondences between the STAIRS paragraph or formatted field names and the CCL field labels is loaded and scanned against the field label list given in the format parameter. The appropriate conversion takes place, and the input string is converted into:

```
..BROWSE <search statement number><format>
```

and then processed as a normal STAIRS command. The DLN005 module has been modified to accept a document range and to allow the processing of the record increment number (skinning list).

E13 - PRINT Command (SCS504,DLN010,DLN005)

The PRINT command is used to have retrieved documents printed on the offline printer or on a private print queue.

The abbreviated form P is also valid input.

The input is not positional. If a parameter should be entered more than once, the last value entered is held to be valid.

The command format is as follows:

Command	Parameter	Default value
P[RINT]	[S = gn]	last query
	[;R = n [TO m]]	1 TO 50
	[;I = k]	1
	[;F = p1;p2;...]	
	or	ALL
	[;Fn]	
	[;D = OFFLINE]	
	or	D = OFFLINE
	[DISK = prtq]	

Where:

- gn = query number
- n = first document number in the list to be displayed
- m = last document number in the list to be displayed
- k = increment document number (for skimming list)
- pn = field label (= STAIRS paragraph or formatted field)
- Fn = predefined format
 - where "n" is a number ranging from 1 to 23.
 - These numbers are in correspondence with the formats "D" to "Z" that can be defined using the DLNPCDEF macro of STAIRS/VS-TLS.
- D = OFFLINE asks for offline printing of documents
- DISK = prtq asks for printing of documents on private data set, identified as "prtq" in DCT

The PRINT command is processed by the SCS504 module.

The search statement number, record number and record increment number are all stored in SCSTWA, while the format is moved into a command area.

The table giving the correspondences between the STAIRS paragraph or formatted field names and the CCL field labels is loaded and scanned against the field label list given in the format parameter. The appropriate conversion takes place, and the input string is converted into:

```
..BROWSE <search statement number><format>
```

and then processed as a normal STAIRS command. The DLN005 module has been modified to mask the output and to simulate a "..PRINT" as input. The module DLN010 has been modified to accept a document range and to allow the processing of the record increment number (skinning list).

Field Labels

(quoted from A.E. Negus and A.E. Snowden)

AU author
TI title
SO source
CC classification code (other specialised codes
are generally database dependent)
CT controlled term (i.e. thesaurus term)
UT uncontrolled term (i.e. free indexing)
FT free text term
LA language
NR report number
CS corporate source
AF author affiliation
NP patent number
CP patent country
CY publication country
JT journal title
JA journal abbreviation
CO CODEN
SS ISSN
SB ISBN
AB abstract

NA abstract number
ND document number
NC contract number
ED computer entry date
PD publication entry date
PY publication year
DT document type
RT reference
RA referenced author
RI referenced inventor
RJ referenced journal
RP referenced patent
CR Chemical Abstracts Registry Number
WL Wiswesser line notation

PLANTAS PUEBLO D.
SERVICIO TIPOGRAFICO
DEL CRUCE

