2.7.80 [illegible]

```
DDDDDDDD          BBBBBBBBBB
DDDDDDDDD         BBBBBBBBBBB
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        BBBBBBBBBBB
DDDDDDDD          BBBBBBBBBB
```

Common Command Language

Implementation on STAIRS/VS - TLS

Release 1

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
Italy

Tel. +39 50 45245
Telex 500371 CNUCE

June 1980

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=CNUCE,
//              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 CNUCE. 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
3.	SCS.TEST.PLILIB	PLILIB to compile SCS modules

4.	SCS.TEST.SOURCE	SOURCE modified for SCS and ad hoc modules Input cards for message file generation Input cards for CCLHELP file generation Input cards for map generation Examples of CICS tables. EURO Database DBCB VIEPARMS
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	Update cards to SOURCE and MACLIB for SCS Update cards to CICS and STAIRS source for TTY support. Examples of procedures Examples of procedures
7.	SCS.HELP.MESSAGE	CCL Help messages file (ISAM)
8.	TNDXEURO	
9.	TEXTEURO	
10.	INVTEURO	
11.	DICTEURO	

Note that the distributed LOADLIB has the TRACE=YES option; this considerably impacts performance and is therefore not suitable in a production environment.
Therefore, if TRACE=NO is desired, the modules DLM020, VIE800, and VIE900 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 IEBISAM utility. The other files have been produced using a standard IEBGENER IBM utility.

The suggested space allocations for these 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 tape must be copied;
3. The new PPT, PCT, FCT and DCT 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 e 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 member CCLHELP of SCS.TEST.SOURCE;
5. The DBCB and PFFILE for the EURO database must be generated and the relative entries must be added to the UREG records. The appropriate cards are given in member EURODBCB of SCS.TEST.SOURCE.
6. In order to have TTY support, the DFHTCCLC, DFHTCEXT, DFHTCRN, and DFHTCTWX members in CICS.SOURCE must be updated using the corresponding members in the SCS.TEST.CARDS and the DFHTCP 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 are different 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 and CCLHFLP DD cards 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
SCSPGISP EQU   X'01'                  INPUT SIM. FOR PAGE CMD          CCL1
*SCSFFIND 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 DASE 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
SCSDSPLY 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
SCWADR0A 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                      BASE & THES. AREA AVAIL. POS.     CCL1
SCWABTNM DS    CL160                  BASE & THES NAME AREA             CCL1
         ORG   SCWABTNM                                                 CCL1
SCWATSRC DS    CL160                  T.S. QUERY RECORD                 CCL1
CCLATFLD DS    F                      ADDRESS OF F.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
CCLLF1   DS    H                      LENGTH OF AREA BASED ON CCLADDR1
CCLLF2   DS    H                      LENGTH OF AREA BASED ON CCLADDR2
CCLLF3   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
SCWASQNO 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 SCWADBNM 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 & THES. AREA AVAIL. POS.CCL1
   3 SCWABTNM CHAR (160),          /* BASE & THES 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 STATAS      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)         /* ADRRESS 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);                 /*                               CCL1
DCL 1 SCWADPLY_WORK                /* DISPLAY VARIABLES             CCL1 */
  BASED(SCWAPTR4),                 /*                               CCL1
      2 SCWADR0A 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 */
SCSPGISP 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 */
SCSPRRN  BIT(8) INIT('01000000') STATIC,/*  REMOTE PRINTING  CCL1 */
SCSPROF  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 define the correspondence 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 | Field |            | Low | Up  | Paragraph   |
|        | level   | label |            | p.c.| p.c.|    code     |
|________|_________|_______|____________|_____|_____|_____________|
| (=F)   | (=255)  | blank | 'FORMFILD' |     |     |             |
|________|_________|_______|____________|_____|_____|_____________|
 <- 1 -> <-  1  -> <- 2 -> <--  8   --> < 2 > < 2 > <--  4    -->  bytes

 Entry (first type=formatted-field).
|--------|---------|-------|------------|-----|-----|-----|------|
| Type   | Privacy | Field |STAIRS name | LL  | Pos | FT  | Par. |
| (=F)   | level   | label |   (FFFIELD)|     |     |     | code |
|________|_________|_______|____________|_____|_____|_____|______|
 <- 1 -> <-  1  -> <- 2 -> <--  8   --> < 2 > < 2 > < 1 > <- 3 ->  bytes

 Entry (second type=paragraph).
|--------|---------|-------|--------------------------------------|
| Type   | Privacy | Field |    STAIRS  name                      |
| (=P)   | level   | label |        (paragraph name)              |
|________|_________|_______|______________________________________|
 <- 1 -> <-  1  -> <- 2 -> <---            16                 --->  bytes

 Last entry.
|-----------------------------------------------------------------|
|'* STOPPER TABLE'                                                |
|_________________________________________________________________|
 <-------                     20                        --------->  bytes
```

In order to build this table the program CCL01 must be executed according to the example given in the member CCLTAB of SCS.TEST.CARDS.

The format of 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 'FORMFILD'. This range must be at least equal to (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 parameter.

C. The general philosophy behind the modifications

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 scanned for all parameters. 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 have been 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 for:

- 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, TCTTWX)

  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 with just 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' will be converted to X'1526'.

- System prompting

  CCL requires that the system is ready for input when the following 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 implies an update to the DFHTCCLC member of the CICS.SOURCE and a new generation of the DFHTCP module.

D2 - Input conversion (DLN020)

The CCL dialog is essentially non-conversational, i.e. each 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)

In certain cases, however, the input must not be interpreted as a command because the system is asking for a specific parameter (e.g. the Database name, the password, the thesaurus name).

In these cases, the input conversion described in previous paragraphs 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 keeping with the STAIRS philosophy, which in certain cases accepts the input ..OFF in order to exit from an endless question-answer loop.

D4 - SCS command definitions (DLN020, DLN010)

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

In the SCS environment, for each input, the SCSCMRCV switch is turned on 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 module in 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 are 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 one of the two files, the standard files are read.

This enables the four SCS files which contain all the modified messages to be defined without the necessity of duplicating the original files.

D7 - Map Selection (VIE900)

The maps are invoked by the modules with standard names, i.e. VIEnnnmm, nnn is the module number and mm is the number of the map. In the SCS environment the map name is changed to SCSnnnmm. If at least one of the SCSSYSHP or

SCSCMSHP flags are on, the name of the map is changed to SCHnnnmm. 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

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, optionally

followed by the user password and name. (However, 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 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 swiched 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 accesses automatically to the EURO data base (which is not associated to a thesaurus) but is completely unaware of this.

At this point, sign on procedure is complete and the user can either issue a command, or just press the "enter" key, thus passing the control to the VIE804 module and obtaining a map which displays the permitted commands.

E2 - BASE Command (SCS501, VIE801)

BASE select the database (name or number) 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.

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.

M3 - 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) - 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 term 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 brackes 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 search history (display of all preceding queries)

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 term.

E3.1.1 - Prefix.

- A prefix is divided into a field (or data element) label and 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. Theretore, a search statement may be qualified using prefix and suffix 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 <f11> = butterfly AND <f11> = daisy
  is equivalent to:
    FIND <f11> = (butterfly AND daisy)
    or:
    FIND (butterfly AND daisy)/<f11>
```

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 on masked term.

Example:

FIND compu$

will retrieve computational, computer, computing, etc.

E3.4 - Syntax limitations

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

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

is coverted 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 an 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.

B4 - 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 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 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 (skimming list).

E5 - 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 = qn]	last query
	[;R = n [TO m]]	1 TO 50
	[;I = k]	1
	[;P = p1;p2;...]	
	or	ALL
	[;Fn]	
	[;D = OFFLINE]	
	or	D = OFFLINE
	[DISK = prtq]	

```
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 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 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 fields 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 (skimming list).

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 ordering 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 SCS507 module. The appropriate values are put in SCSTWA or DLNTWA.

E7 - 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 afterward a reference with T= is made in FIND command, it will be relative to this display.

When DLN013 is going to write results, if the DISPLAY function is active, returns to SCS506.
The SCS506 module reads temporary storage records written by DLN047, release them, and writes a "record 0" and one record for each term found.

At this point, in both cases, SCS506 shows a map where terms are sequentially numbered, and which contains relators and descriptors in case of CT display, and number of occurrences and words in case of FT display.

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

E8 - OWN Command (DLN002, DLN010)

The OWN command allows use of STAIRS/VS-TLS own commands. 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 the DLN002 or 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 - 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.

E10 - HELP Command (SCS505)

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

Also H and ? are 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.

SCS505 reads messages from CCLHELP file, and displays them. In case of wrong input,

HELP HELP

is assumed.

If the output consists of more than one page, the user can move backward and forward using BACK, MORE and PAGE commands.

E11 - BACK Command (DLN020)

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

Also B is 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   in case of 3270

and

doc-n in case of TTY

E12 - MORE Command (DLN020)

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

Also M is accepted as valid input.

The command format is:

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

Where:
number   indicates how many pages the user want to move forward

The input is converted into:

p-n   in case of 3270

and

doc-n in case of TTY

A "carriage return" is taken as:

MORE 1

E13 - PAGE Command (DLN020)

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

Also PA is 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
```

F - Command Implementation

F1 - Sign on

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, optionally followed by the user password and name. (However, if STAIRS is entered, name and password are mandatory). When one of these parameters is entered, the appropriate transaction is initiated. If no parameter is entered, CCL is assumed by default.

When sign on procedure is complete, the user can either issue a command, or just press the "enter" key, thus obtaining a map which displays the permitted commands.

E2 - BASE Command

BASE selects the database (name or number) 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.

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.

F3 - FIND Command

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) - 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 term 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 brackes 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 search history (display of all preceding queries)

F3.1 - 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 term.

F3.1.1 - Prefix.

- A prefix is divided into a field (or data element) label and 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>

F3.1.2 - 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 prefix and suffix only if the relational connector '=' or a thesaurus relator are used.

3.1.3 - 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
```

3.2 - 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:

F3.1.3 - Prefix and parentheses

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

```
  Examples:
      FIND <f11> = butterfly AND <f11> = daisy
    is equivalent to:
      FIND <f11> = (butterfly AND daisy)
      or:
      FIND (butterfly AND daisy)/<f11>

  and:
      FIND AU = (schultz OR parker OR hart)
    is equivalent to:
      FIND AU = (OR schultz;parker;hart)
      or:
      FIND (OR schultz;parker;hart)/AU
```

F3.2 - 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>

F3.3 - Truncation

The character masking symbol ($) is used to search on masked term.

Example:

```
FIND compu$
```

will retrieve computational, computer, computing, etc.

F3.4 - Syntax limitations

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

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

is coverted into:

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

F4 - SHOW Command

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 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 formats
"D" to "Z" that can be defined using the
DLNPCDEF macro of STAIRS/VS-TLS.

F5 - PRINT Command

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:
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 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 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
```

F6 - DEFINE Command

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

F7 - DISPLAY Command

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  [TO term2]           |
|           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
TO <term2> identifies the last free text word to be displayed
<number> = a number indicating any display already made. If afterward a reference with T= is made in FIND command, it will be relative to this display.

? asks for display of the last successfull display. If afterward a reference with T= is made in FIND command, it will be relative to this display.

A map is showed containing:

a) - relator and words in case of CT display;

b) - # of occurrences and words in case of FT display

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

F8 - OWN Command

The OWN command allows use of STAIRS/VS-TLS own commands.

The command format is:

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

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

F9 - STOP Command

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

The command format is:

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

F10 - HELP Command

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

Also H and ? are 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 the output consists of more than one page, the user can move backward and forward using BACK, MORE and PAGE commands.

F11 - BACK Command

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

Also B is 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

F12 - MORE Command

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

Also M is accepted as valid input.

The command format is:

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

Where:
number   indicates how many pages the user want to move forward

A "carriage return" is taken as:

```
MORE 1
```

F13 - PAGE Command

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

Also PA is 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.