

WIP

DDDDDDDDDD		BBBBBBBBBBBB	
DDDDDDDEDD		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
DD	DD	BB	BB
DDDDDDDDDD		BBBBBBBBBBBB	
DDDDDDDDDD		BBBBBBBBBBBB	

Common Command Language

Support on STAIRS/VS - TLS

Phase 2 - Working Document

Reparto
Basi di dati e sistemi informativi

R. Bartoli
G.A. Romano
O. Signore

CNUCE
Via S. Maria, 36
56100 Pisa

Tel. +39 50 45245
Telex 500371 CNUCE

February 1981

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		BBBBBBBBBBEE	

Common Command Language
 Support on STAIRS/VS - TLS
 Phase 2 - Working Document

Reparto
 Basi di dati e sistemi informativi

R. Bartoli
 G.A. Romano
 O. Signore

CNUCE
 Via S. Maria, 36
 56100 Pisa

 Tel. +39 50 45245
 Telex 500371 CNUCE

February 1981

Table of contents

Foreword	1
A. Project status	2
B. Command Implementation	4
B1. Sign on	5
B2. HELP Command	5
B3. BACK Command	
B4. MORE Command	6
B5. PAGE Command	7
B6. NEWS Command	7
B7. INFO Command	8
B8. DEFINE Command	9
B9. DELETE Command	10
B10. BASE Command	10
B11. STOP Command	12
B12. FIND Command	13
B13. DISPLAY Command	15
B14. SAVE Command	16
B15. SHOW Command	17
B16. PRINT Command	18
B17. OWN Command	20
B18. ..CCL Command	22
B19. Miscellaneous	
C. Conclusions	23
References	27

Foreword

The aim of this document is to provide a basis for discussion when defining the content of Phase 2 of the implementation of the Common Command Language (CCL) on STAIRS/VS - TLS.

The document will contain a brief description of the present state of the implementation, a review of the commands as they are currently implemented, proposals for future developments and additions to the original A.E. Negus proposal which have been suggested by an Italian user. Finally, maintenance problems and the extent to which the manufacturer, the other hosts and the European Commission should become involved are discussed.

A. Project status

The most important stages of the project are outlined below:

- December 1979: first CCL implementation ready ([CNUCE79]).
 - 19 December 1979: meeting at CNUCE with A.E. Negus and IBM.
 - 20 December 1979: European Commission is advised of availability of first implementation of CCL.
 - 22 January 1980: meeting at SCICON, some amendments to the implementation are agreed.
 - 29 January 1980: meeting in Luxembourg, presentation of CNUCE implementation to interested hosts. It is agreed that these hosts should receive a delivery tape containing the amendments discussed in the meeting of 22.1.80.
 - 03 March 1980: delivery of first implementation tape containing the amendments agreed 29 January at Luxembourg, and additional enhancements.
 - 10 April 1980: meeting in Luxembourg to discuss hosts' experiences with the implementation distributed, and the additional developments proposed by CNUCE.
- IAEA presents a report on its installation experience.
- The use of an inverted string technique in the implementation of the FIND command was rejected, at

least for the moment, due to the problems involved in the necessary restructuring of databases. After discussion, an implementation in which it was not possible to mix SEARCH and SELECT type queries was agreed.

- 22/23 April 1980: workshop at CNUCE with IBM representatives. The content of the CCL implementation, to be accepted by the EC in fulfilment of the contract was agreed ([CCL-0001-1]).
- 23 May 1980: delivery of CCL Release 0 tape ([CNUCE80a]).
- 07 July 1980: delivery of CCL Release 1.0 tape ([CNUCE80b]).
- 28 August 1980: distribution of reporting on first 6 errors discovered on CCL Release 1.0, operational at CNUCE.
- 14 November 1980: the European Commission is officially informed of the completion of the implementation.
- 20 January 1981: an Italian firm, after having tested various CCL software, selects the CNUCE implementation on STAIRS/VS-TLS.
- 10 February 1981: meeting in Luxembourg to agree the content of Phase 2 of the implementation.

B. Further Command Implementation

In the following paragraphs, all the CCL commands are described, one by one. The present state of implementation and the final syntax expected is given for each command.

The future developments suggested are the result of a very general examination of the situation and are intended to be used as a basis for discussion during the definition of the content of Phase 2.

In the Miscellaneous section, some of the suggestions made by the Italian firm, which has selected the CCL implementation on STAIRS/VS - TIS in order to provide a service on a national scale, are presented.

B1. Sign on

Present implementation:

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

Further developments:

A restart on the database will be made possible if it is decided to support the BUT parameter on the STOP command.

B2. HELP Command

Present implementation:

```
|=====|
| Command          Parameter          Default value |
|=====|
| H[ELP] or ?     [ command ]         HELP
|-----|
```

Further developments:

None.

B3. BACK Command

Present implementation:

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

Further developments:

None.

B4. MORE Command

Present implementation:

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

Further developments:

None.

B5. PAGE Command

Present implementation:

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

Further developments:

None.

B6. NEWS Command

Present implementation:

None.

Future developments:

Command	Parameter	Default value
NEWS	?	

May be implemented by accessing a specific "scratch pad" file.

B7. INFO Command

Present implementation:

None.

Future developments:

Command	Parameter	Default value
INFO	EURONET	
	or COST	
	or SCHEDULE	
	or USERS	
	or STATUS	

May be implemented by adding specific entries to the HELP messages file or by setting up an ad hoc file.

B8. DEFINE Command

Present implementation:

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

Further developments:

Command	Parameter	Default value
DE[FINE]	[FN = <number>]	last + 1
	[;FORMAT = fl1,fl2,...]	none

If the FORMAT parameter is omitted, a prompting dialog takes place. The format defined is checked against the CCL correspondence table and is then stored in a file. Consequently, SHOW and PRINT will be modified to take output format from this file instead of from the DLNFCDEF table.

OP: $F_n = fl_1; fl_2; fl_3$
 SYS OK .. but the order is the system order

B9. DELETE Command

Present implementation:

None.

Future developments:

Command	Parameter	Default value
DEL[ETE]	[S = n [TC m]]	none
	[;P = n [TC m]]	none
	[;SAVE = < profile name >]	none

The CCL queries, print requests and saved profiles are deleted.

B10. BASE Command

Present implementation:

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

Further developments:

Command	Parameter	Default
BAS[E]	[;ED = date1 TO date2] [;NOFIND]	none

a) - ED = date1 TO date2

In module VIE801:

```
..SELECT ALL ED WI date1,date2
```

is simulated as input at "signon complete time".

The display of the result may be altered for this first input in module DLN025.

Then, with a FIND "search-type", the query is modified to:

```
..SEARCH 1 AND <query>
```

With a FIND "select-type":

- if the query contains a backreference it is not modified;

- if the query does not contain a backreference, it is transformed into:

```
..SELECT 1 <query>
```

The implementation of this parameter does not seem particularly useful or meaningful as it would give very poor performances unless an inverted string

*generare un
altro TMSX
file wlx
l'Ed. date-
le prime select
venite fatte nel
nuovo.*

technique is adopted to effect SELECT type queries.

b) - NOFIND

If the NOFIND parameter is entered, a simple control of the CCL syntax is made, and the query is not executed.

B11. STOP Command

Present implementation:

Command	Parameter	Default value
STOP		

Further developments:

Command	Parameter	Default value
STOP	[BUT]	none

If the BUT parameter is entered, the input is converted into

..OFF CCNT

and a database restart is possible. The sign on implementation will be modified accordingly.

B12. FIND Command

Present implementation:

Command	Parameter	Default value
F[IND]	identifier [operator identifier] ...	
OR		
F[IND]	?	

Further developments:

Command	Parameter	Default value
F[IND]	SAVE = <profile name>	
	[;NOFIND]	

Queries saved with the SAVE command and identified by <profile name> are submitted to the FIND stack builder program and executed, with suitable modifications being made to any backreferences. If the NOFIND parameter is entered the queries will not be executed.

Further extensions:

Reference to terms previously shown using the DISPLAY command can be made explicitly and dynamically; thus, in a single query, reference can be made not only to the "current display" but also to different displays.

For example, a statement of the type:

```
FIND T=3.5 AND (T=8.12 TO 15)
```

would put the fifth term of the third display in AND with the logical OR of the terms from 12 to 15 of display number 8.

Note: At the present, there are two restrictions:

- i) - the max. length of the query is limited;
- ii) - SEARCH and SELECT type queries can not be combined.

The first of these two restrictions is caused by the fact that STAIRS does not accept queries longer than 234 bytes. Consequently, queries of the following type cannot be made on a wide range of terms.

```
FIND T= <n> TO <m>
```

In the present implementation, the maximum length has been set at 138 bytes, this could be carried without any difficulty to over 234 bytes.

If the second restriction is to be overcome while still maintaining an acceptable performance level then the data base structure must be modified by introducing an ad hoc paragraph for the formatted fields. In this way, the SELECT query would be transformed into SEARCH, as was shown at the meeting of 10 April 1980 at Luxembourg.

This solution would also enable the ED parameter to be supported efficiently.

B13. DISPLAY Command

Present implementation:

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

Further developments:

None.

B14. SAVE Command

Present implementation:

None.

Further developments:

Command	Parameter	Default value
SA[VE]	<profile name>	none

The queries are saved with the display terms which have been specified.

Queries recalled by

FIND SAVE=<profile name>

are re-executed.

B15. SHOW Command

Present implementation:

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

Further developments:

If the output formats can be defined dynamically using the DEFINE command, the PL/I module must be modified to read the formats from a file instead of from the table DLNCDEF.

B16. PRINT Command

Present implementation:

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

Further developments:

Command	Parameter	Default value
P[RINT]	[L = <char string>]	blank
	[;SORT = f11,f12,]	ED (?)
	[;ED = <date>]	none
	[;HIST = YES NO]	NO

a) - L=

this could be supported by writing a suitable print record and by modifying the offline print program.

b) - SORT =

this is only possible on field labels with a linear range (i.e. formatted fields).

c) - TL =

this cannot be supported

d) - ED =

this could be supported by executing a suitable SELECT (or SEARCH if the DB structure is altered as described in FIND).

e) - HIST =

this could be supported by writing an appropriate record on file DLNPRINT and modifying the offline print program. All executed queries will be printed.

In addition, if it becomes possible to define the output formats dynamically using the command DEFINE, then the PL/I module will have to be altered so that the formats are read from a file instead of from the table DLNPCDEF.

B17. OWN Command

Present implementation:

Command	Parameter	Default value
OWN		

Further developments:

Command	Parameter	Default value
OWN	[< STAIRS command >]	none

If OWN is followed by a STAIRS command, this command will be executed and the user will then remain in STAIRS environment.

B18. ..CCL Command

Present implementation:

Command	Parameter	Default value
..CCL		

Further developments:

Command	Parameter	Default value
..CCL	[< CCL command >]	none

If the command ..CCL is followed by a CCL command, this command will be executed and the user will then remain in the CCL environment.

B19. Miscellaneous

Some other proposed functions are briefly outlined:

- a) - At the end of each database session, the cost of the session should be given.

When the command STOP is given, the user should have displayed the costs of the individual database sessions together with the total cost of the entire session.

- b) - It should be possible to access to the accounting for the whole of the period which has not yet been invoiced.

no
P1/R0?

- c) - It should be possible to verify which of the DBs actually available are most relevant for the scope of a particular session (i.e. to make the same query on all the accessible DBs).

Good idea
but
impossible
Time consuming.

- d) - On-line ordering of original documents.

- e) - SDI and batch terminal supports.

- f) - On-line deleting of the original document ordering.

- g) - Off-line printing of non-standard formats.

C. Conclusions

The present implementation has been made so as to support all those CCL features which were not in contrast with the STAIRS basic philosophy (a typical example is the SEARCH - SELECT). Nevertheless, the fact that CCL is command driven while STAIRS follows the "environment" concept (i.e. distinct SEARCH, BROWSE and SELECT environments) has often created serious problems, forcing us to make a number of alterations in certain modules. In addition, STAIRS, probably partly as a consequence of this "environment" philosophy, does not have an authentic Command Driver. The commands are tested and accepted in a number of different modules (DLN002, DLN010, DLN013, DLN025, VIE804). The DLNCMDEF table does not help to resolve this problem as, although it makes it possible to define other commands in addition to the STAIRS commands, STAIRS commands are still always accepted. This fact has also implied a series of minor, interspersed modifications in some of the "environment" drivers to avoid STAIRS commands being accepted in the CCL environment with unpredictable results. Consequently, the present implementation consists of approximately 6000 PL/I, 1600 Assembler and 800 Macro assembler statements. The Assembler alterations, which from the maintenance viewpoint are the most heavy, could be

eliminated if certain small enhancements were added to STAIRS-TLS. (i.e. a skimming list and the possibility of entering as valid input the ..SEARCH command together with the query or the ROOT). If the manufacturer was willing to include these suggested improvements, the hosts would be much more ready to install the implementation as a major, if not total compatibility, with successive releases would be ensured.

A number of possible solutions are suggested with regard to the eventual involvement of the Hosts.

- a) - any enhancements of STAIRS/VS - TLS which have already been developed and tested by the Hosts, could be included in the final implementation. (Some collaboration between the various Hosts on this point would be necessary).
- b) - the development of the additional functions proposed by Phase 2 could be equally distributed among the future users. In this case, it could be necessary to establish a center for coordination and testing.

During the joint meeting between the "Programme Committee" of the User Forum and the "Standing Committee" of EHOOG held in Luxembourg, 16th December 1980, some considerations were made on CCL by both users and hosts:

- "At present, the hosts offering the CCL only give training courses in the most specific retrieval language, and the availability of the CCL is only mentioned briefly. This attitude of the hosts should change and CCL should be seen as a major selling argument."

- "If the CCL is implemented by one host the commands should be consistent with others. Either complete new software implementation or an interface (mapping) are needed.

If mapping of the CCL is chosen by different hosts on compatible mainframes one consistent interface should be guaranteed even though some shortcomings cannot be avoided.

It was emphasized that two slightly different CCL versions are worse than two completely different systems." (+)

- "The seed money provided by the CEC for developing and implementing CCL was only 1/3 to 1/5 of the real costs. Therefore the CEC was invited to provide more financial

(+) It might be advisable to create a group with the specific responsibility for the analysis, evaluation and coordination of these different versions. The European Commission should be willing to provide financial support for a group of this type.

support to those hosts which commit themselves to further developments of the CCL."

Acknowledgments

A warm thankyou is given to S. Lippi (IBM, Italy) for helpful discussions on the feasibility of further implementations.

References

- [CCL-0001-1] Euronet/CCL Project team: EURONET/CCL implementation with STAIRS/VS - Document number CCL-0001-1
April 23rd, 1980
- [CNUCE79] Common Command Language - Implementation on STAIRS/VS - (December 1979)
- [CNUCE80a] Bartoli R., Lippi S., Romano G.A., Signore O.: Common Command Language - Implementation on STAIRS/VS - Release 0;
CNUCE Report n. 174, Luglio 1980
- [CNUCE80b] Bartoli R., Lippi S., Romano G.A., Signore O.: Common Command Language - Implementation on STAIRS/VS - TLS - Release 1;
July 1980
- [CNUCE80c] Bartoli R., Lippi S., Romano G.A., Signore O.: Common Command Language - Implementation on STAIRS/VS - TLS - Implementation Description (Release 1.1)
October 1980
- [CNUCE80d] Bartoli R., Lippi S., Romano G.A., Signore O.: Common Command Language - Implementation on STAIRS/VS - TLS - User Guide (Release 1.1)
October 1980
- [MODDER80] Modderkolk, Albert A.: Standardized Command Set - An Overview of the Modification to the IBM Products STAIRS/VS and STAIRS/VS - TLS
(January 28th, 1980)
- [NEGUS77] Negus A.E.: Euronet Guidelines: Standard Commands for Retrieval Systems.
London, The Institution of Electrical Engineers, 1977, 66pp.
- [NEGUS79] Negus A.E., Snowden A.E.: EURONET-DIANE User's Guide (Common Command Language) - Draft for Comment
1979, Scicon Consultancy International Ltd.

- [SCHREI80] Schreiber F.A., Difilippo C., Zagolin M.: Un preprocessor per l' interfacciamento di sistemi eterogenei di information retrieval su una rete di calcolatori
Atti del Congresso Annuale AICA '80, 29/31 Ottobre 1980, Bologna, pp. 1245-1263
- [STAIRSgi] Storage and Information Retrieval System/Virtual Storage (STAIRS/VS) - General Information
IBM GH12-5114
- [STAIRSog] Storage and Information Retrieval System/Virtual Storage (STAIRS/VS) - Operations Guide
IBM SH12-5500
- [STAIRSprm] Storage and Information Retrieval System/Virtual Storage (STAIRS/VS) - Program Reference Manual
IBM SH12-5400
- [TlSgi] Storage and Information Retrieval System/Virtual Storage - Thesaurus and Linguistic Integrated System - General Information Manual
IBM GH12-5123
- [TlSor] Storage and Information Retrieval System/Virtual Storage - Thesaurus and Linguistic Integrated System: Online Retrieval - Program Description and Operations Manual
IBM SH12-5318
- [TlStg] Storage and Information Retrieval System/Virtual Storage - Thesaurus and Linguistic Integrated System: Thesaurus Generator - Program Description and Operations Manual
IBM SH12-5319

