

→PEHQ

RCSL No: 43-GL11149

Edition: March 1981

Author: Uffe Harksen

Title:

NCTH
User's Guide

Keywords:

PAXNET, REAL TIME PASCAL, RC3502.

Abstract:

This manual contains a description of the NCTH-module in PAXNET, and a description of the implemented MML-like command language.

(46 printed pages)

Copyright © 1981, A/S Regnecentralen af 1979
RC Computer A/S
Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

1.	N C T H.....	2
1.1	Introduction.....	2
1.2	The NCP-interface.....	2
1.3	COMINT.....	2
1.3.1	INCHAR.....	2
1.3.2	LEXAN.....	2
1.3.3	SYNTAN.....	3
1.3.4	SEMAN.....	3
1.3.5	COMMAND.....	3
1.3.6	Printoutfacilities.....	3
1.3.7	Interface to NCP- and Formatter-module.....	3
1.4	FORMATTER.....	5
1.4.1	The lcp-mess-format.....	5
1.4.2	The Comint-mess-format.....	6
2.	N C T H - M L L	7
2.1	Introduction.....	7
2.2	The lexical symbols.....	7
2.2.1	The syntactical elements.....	7
2.2.1.1	<ident>.....	7
2.2.1.2	<number>.....	7
2.2.1.3	<txt-str>.....	8
2.2.1.4	<symb-name>.....	8
2.2.1.5	<separator>.....	9
2.2.1.6	<def-num>.....	9
2.2.1.7	<pref-num>.....	10
2.2.1.8	<prefix>.....	10
2.2.1.9	<equal>.....	10
2.2.1.10	<d-amp>.....	10
2.2.1.11	<s-amp>.....	11
2.2.1.12	<star>.....	11
2.2.1.13	<comma>.....	11
2.2.1.14	<colon>.....	11
2.2.1.15	<terminator>.....	11
2.2.1.16	<semicolon>.....	12
2.2.1.17	<hyphen>.....	12
2.2.1.18	<spaces>.....	12
2.2.1.19	<bin-prefix>.....	12
2.2.1.20	<oct-prefix>.....	12
2.2.1.21	<dec-prefix>.....	13
2.2.1.22	<hex-prefix>.....	13
2.2.1.23	<digit>.....	13
2.2.1.24	<hex-dig>.....	13
2.2.1.25	<letter>.....	13
2.2.1.26	<spec. char>.....	14
2.2.2	The lexical rules.....	14
2.2.2.1	Rule 1.....	14
2.2.2.2	Rule 2.....	14
2.2.2.3	Rule 3.....	15
2.3	The command-syntax.....	16
2.3.1	The syntactical elements.....	16
2.3.1.1	<info-unit>.....	16
2.3.1.2	<argument>.....	16
2.3.1.3	<sub-value>.....	16
2.3.1.4	<value>.....	17
2.3.1.5	<param>.....	17
2.3.1.6	<block>.....	17
2.3.1.7	<sub-com>.....	17


```

lcpidenttype = PACKED RECORD
    i      : bit;
    id     : 0..32767;
END;
sptypetype = PACKED RECORD
    req_ans      : req_ans_type;
    mess_ev      : mess_ev_type;
    ncp_control  : ncp_cntr_type;
    rep_func     : rep_func_type;
    rej_func     : rej_func_type;
    stat_cntr    : stat_cntr_type;
    ?,?         : bit;
END;

spstatusbit = ( spstatus15,
    sp_status_14,
    sp_status_13,
    sp_status_12,
    sp_status_11,
    sp_status_10,
    sp_status_9 ,
    rep_res_lack,
    no_free_res,
    dupl_func,
    out_of_range,
    illegal_state,
    data_error,
    lcp_unknown,
    ill_lcp_oper,
    chain );

timedigit = 0..15; reqanstype = (req,ans);
messevtype = (mess,event);
ncpcntrtype = (nonpcntr,ncpcntr);
repfunctype = (startrep,stoprep);
rejfunctype = (noreject,reject);
statcntrtype = (noresetstat,resetstat);

basis_type = (lcp_cntr,      (* lcp control operation  *)
    lcp_sense,      (* lcp sense operation  *)
    lcp_get_stat,  (* lcp get statistics oper.*)
    lcp_event );  (* lcp event operation  *)

```

1.4 FORMATTER.

1.4

Messages to formatter must satisfy SP-format described in (3) sec. 2.1.2 and 2.2.1.

The present implemented version only have two printout-formats: Lcp-message-format and Comint-message-format.

The usage of optional facilities is described in sec. 3.3.

1.4.1 The lcp-mess-format.

1.4.1

The Lcp-message-format is separated in two parts:

1. Message header:

```

LCP-MESSAGE
SENDER_ID:      ( Printed in decimal )
SEQ NO:         (      do.      )
SP TYPE:        ( Printed in binary  )
LCP OPER:       ( Printed in decimal )
STATUS:         ( Printed in binary  )
TIME:           ( Printed bitwise in hex. )
BYTECOUNT:    ( Printed in decimal )

```

The message header may optionally be suppressed except for the first two lines.

2. Datapart

The data may be printed in two optional formats. As the header, the datapart may be suppressed.

Vertical printout of spdata-contents:

NO. DEC HEX ASC

Succeeded by <bytecount>-no. of lines containing byteno. and contents in dec ,hex and ASCII-form if in (.0..127.) .

Horizontal printout of spdata-contents:

A sufficient no. of lines of the format:

<byteno> : <bytelist1> <bytelist2>

where

<byteno> is the no of the first byte on this line.

<bytelist1> is a no of bytevalues printed in default-base.

<bytelist2> is the same bytes in ASCII-form.

The last message printed may be kept and repeated an arbitrary no. of times, until the next message arrives.

1.4.2 The Comint-mess-format.

1.4.2

The Comint-message-format is used by internal NCTH-messages exclusively:

line 1: Error no.: <errno.>
2: near position <pos. >
3: <text>

Line 1 is omitted if errno. greater than 49. Line 2 is omitted if pos. less than or equals 0 .

2. N C T H - M L L

2.

Command language.

2.1 Introduction.

2.1

The MML-syntax (Man-Machine-Language) is mainly extracted from (1)-(2). Technically the syntax is based on a few lexical symbols:

Identifiers, numbers, text-strings,
symbolic names and separators.

The syntactical reductions (sec. 2.3) is used by Seman to build up parameter-values delivered to active command-procedures (sec. 3.).

2.2 The lexical symbols.

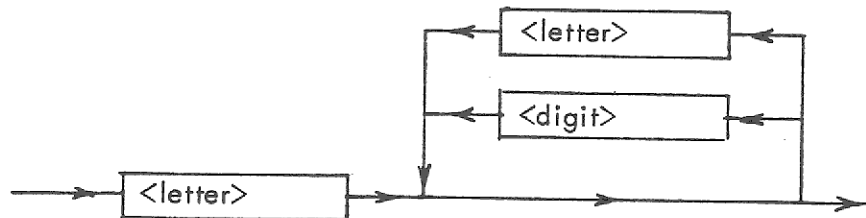
2.2

2.2.1 The syntactical elements.

2.2.1

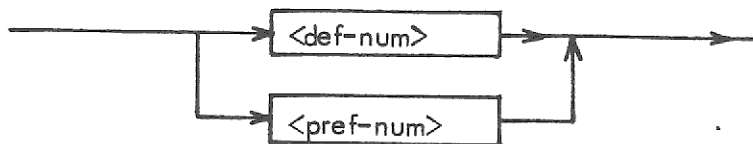
2.2.1.1 <ident>

2.2.1.1



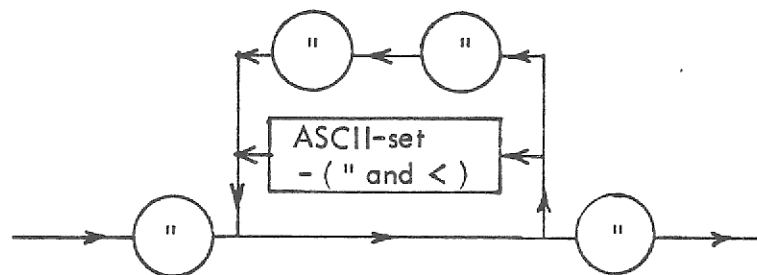
2.2.1.2 <number>

2.2.1.2



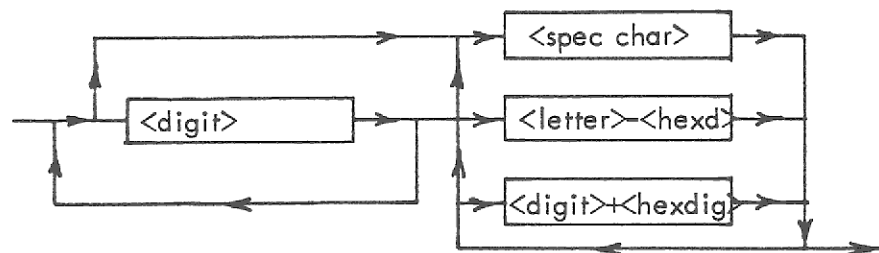
2.2.1.3 <txt-str>

2.2.1.3



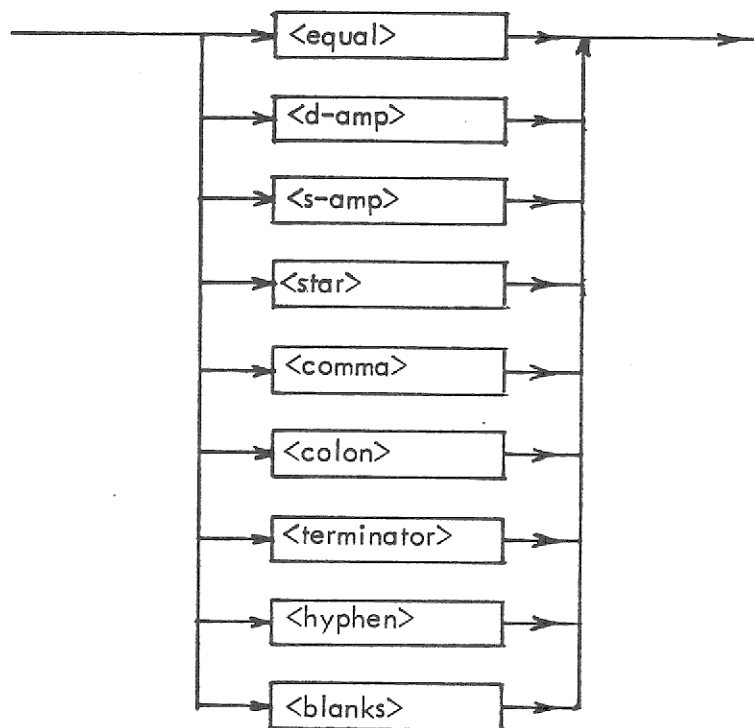
2.2.1.4 <symb-name>

2.2.1.4

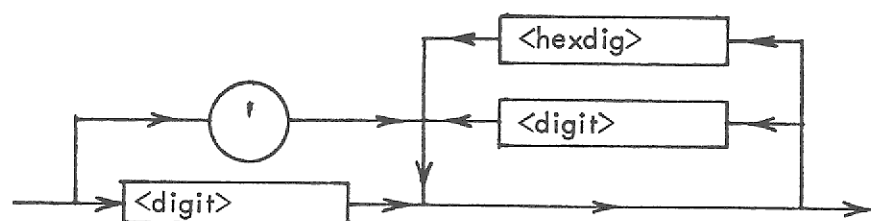


2.2.1.5 <separator>

2.2.1.5

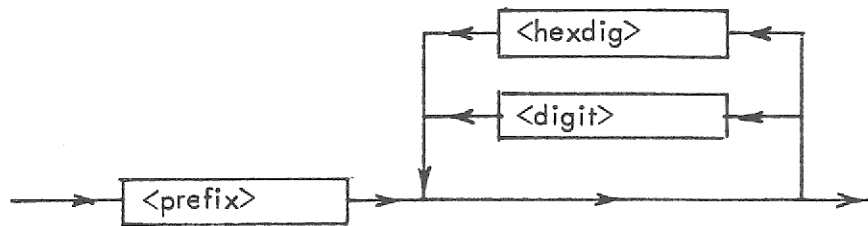
2.2.1.6 <def-num>

2.2.1.6

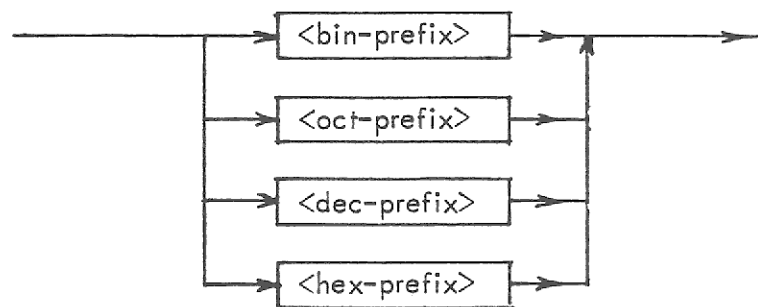


2.2.1.7 <pref-num>

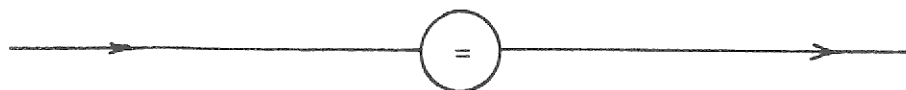
2.2.1.7

2.2.1.8 <prefix>

2.2.1.8

2.2.1.9 <equal>

2.2.1.9

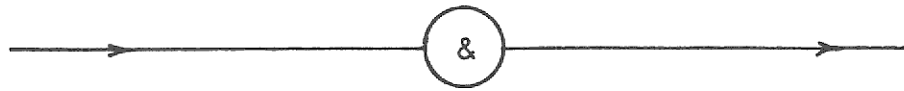
2.2.1.10 <d-amp>

2.2.1.10

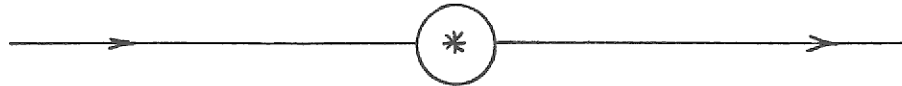


2.2.1.11 <s-amp>

2.2.1.11

2.2.1.12 <star>

2.2.1.12

2.2.1.13 <comma>

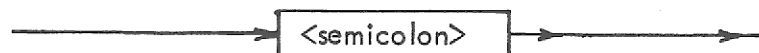
2.2.1.13

2.2.1.14 <colon>

2.2.1.14

2.2.1.15 <terminator>

2.2.1.15

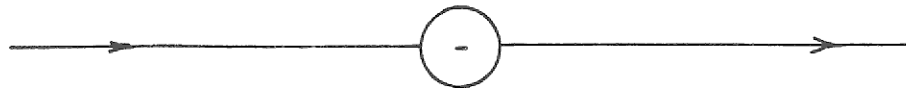


2.2.1.16 <semicolon>

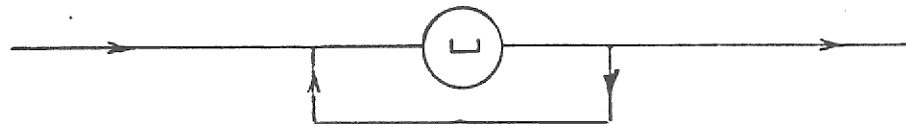
2.2.1.16

2.2.1.17 <hyphen>

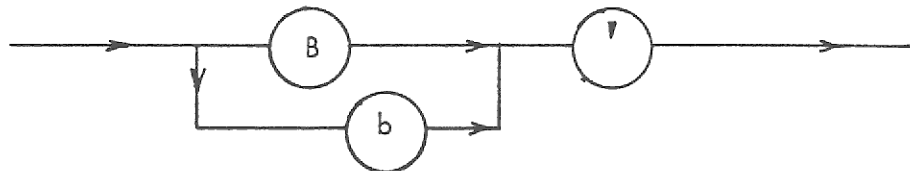
2.2.1.17

2.2.1.18 <spaces>

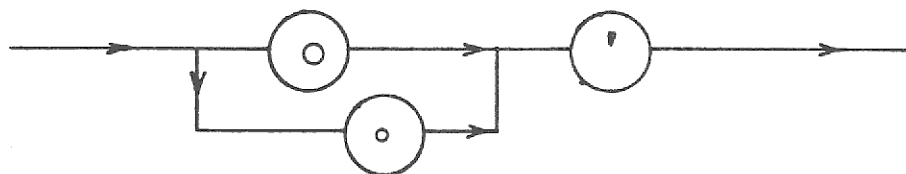
2.2.1.18

2.2.1.19 <bin-prefix>

2.2.1.19

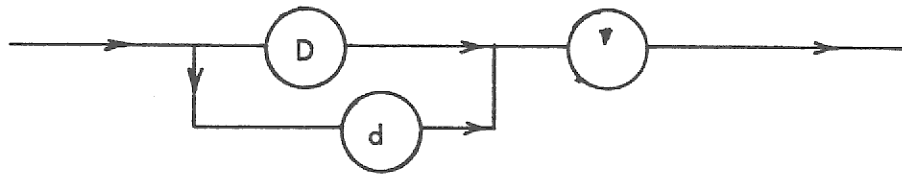
2.2.1.20 <oct-prefix>

2.2.1.20

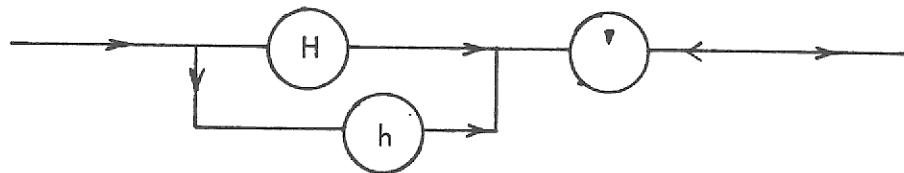


2.2.1.21 <dec-prefix>

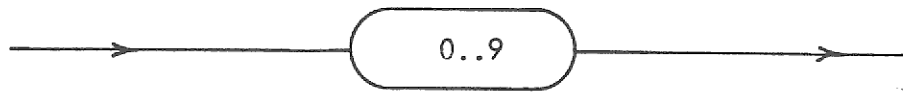
2.2.1.21

2.2.1.22 <hex-prefix>

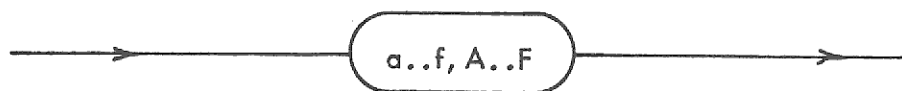
2.2.1.22

2.2.1.23 <digit>

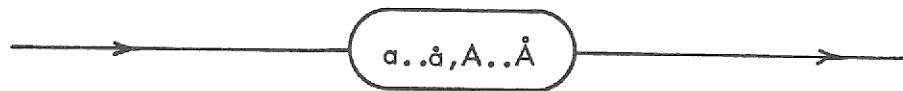
2.2.1.23

2.2.1.24 <hex-dig>

2.2.1.24

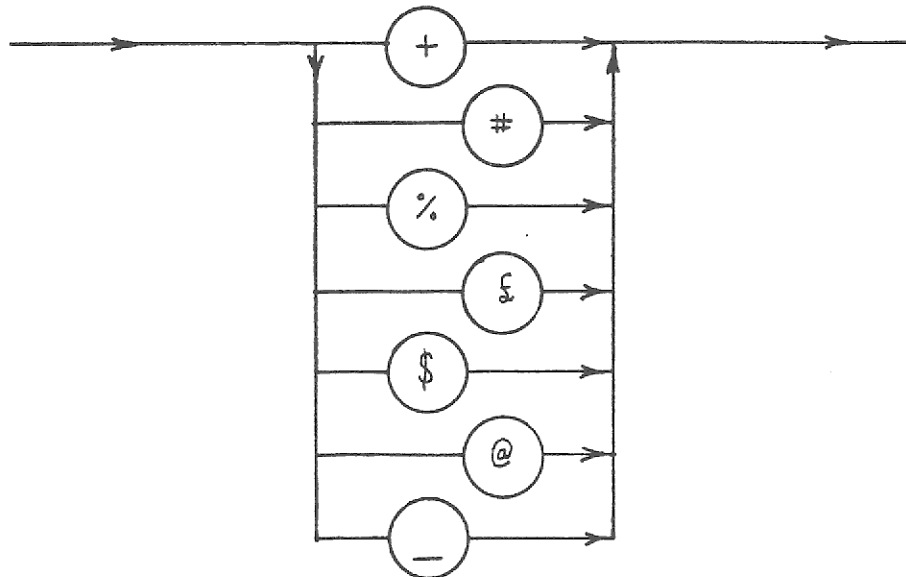
2.2.1.25 <letter>

2.2.1.25



2.2.1.26 <spec. char>

2.2.1.26

2.2.2 The lexical rules

2.2.2

2.2.2.1 Rule 1.

2.2.2.1

A number may only contain digits in the domain $(.0..currentbase-1.)$, where *currentbase* is either *default-* or *prefix-selected* (*currentbase* in $(.2,8,10,16.)$). The maximum decimal number-value is 32765.

2.2.2.2 Rule 2.

2.2.2.2

A *default-number* (*currentbase* = *defaultbase*) not satisfying rule 1 will be interpreted as a symbolic name.

2.2.2.3 Rule 3.

2.2.2.3

Any number of blanks may be inserted between two lexical symbols. They are nonsignificant with respect to the syntax.

2.3 The command-syntax.

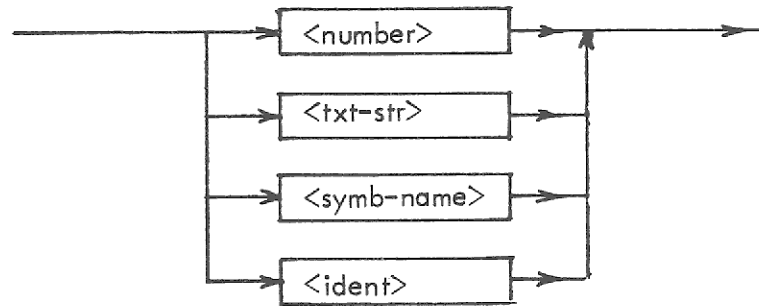
2.3

2.3.1 The syntactical elements.

2.3.1

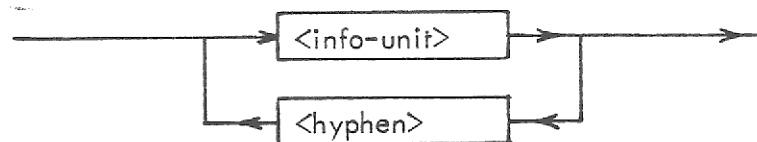
2.3.1.1 <info-unit>

2.3.1.1



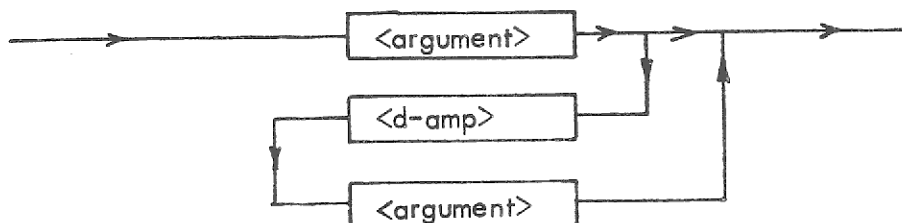
2.3.1.2 <argument>

2.3.1.2



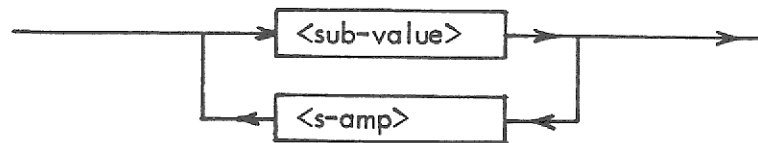
2.3.1.3 <sub-value>

2.3.1.3

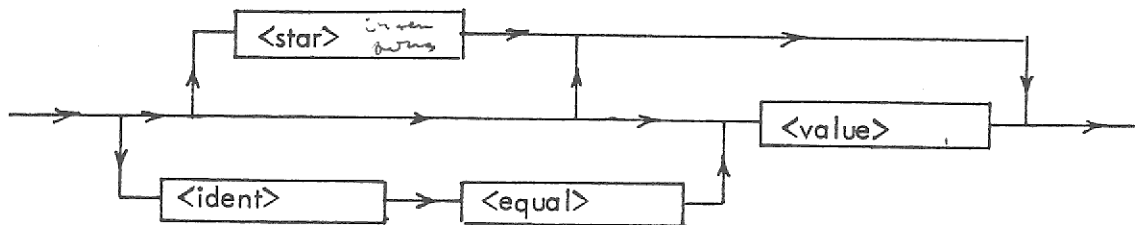


2.3.1.4 <value>

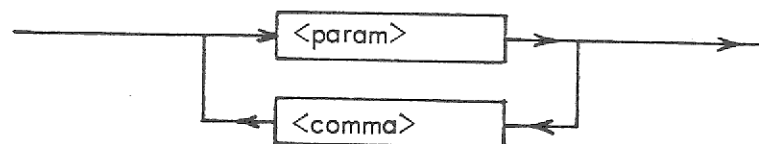
2.3.1.4

2.3.1.5 <param>

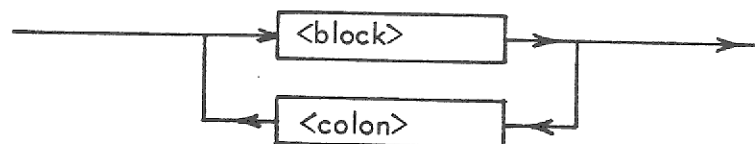
2.3.1.5

2.3.1.6 <block>

2.3.1.6

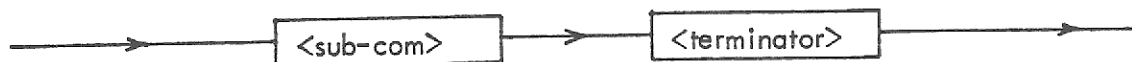
2.3.1.7 <sub-com>

2.3.1.7



2.3.1.8 <command>

2.3.1.8

2.3.2 The syntactical rules.

2.3.2

2.3.2.1 Rule 1.

2.3.2.1

The rest of the line succeeding a <terminator> is skipped. I. e. a comment may be inserted in the command-stream as a semicolon succeeded by as many characters as the linelength allows.

2.4 The semantics.

2.4

2.4.1 The basic types.

2.4.1

Semantically following Info-unit-types are distinguished:

Bytes in (. 0..255 .)

Word in (. 0..32765 .)

Txt = array of ASCII-set excl. (.esc, '<' .)

Symb_name = array of <digit> + <letter>
+ <spec-char>

Command = special identifier only accessible through defined constants.

2.4.2 The semantical rules.

2.4.2

2.4.2.1 Rule 1.

2.4.2.1

An argument may only consist of identical Info-unit-types. A parameter-definition may put an upper limit to the numbers of Info-units in an argument.

2.4.2.2 Rule 2.

2.4.2.2

Text-strings (or symb-names) are treated as arguments consisting of single-character info-units. Two consecutive textstrings (symb-names) are treated as 1 concatenated text-string (symb-name).

2.4.2.3 Rule 3.

2.4.2.3

In NCTH-MML double ampersand (d-amp) is interpreted as a repetition-code.

I. e.:

<arg1> <d-amp> <arg2>

is interpreted as

<arg1><hyphen><arg1><hyphen>...<arg1>

That is : <arg1> repeated <arg2>-no. of times, where <arg2> may only contain a single byte or word less than 256.

2.4.2.4 Rule 4.

2.4.2.4

Let the type of an argument be the type of its info-units. Then a <value> may contain an arbitrary mixture of argument-types depending on parameter-definition. The parameter-definition may also put an upper limit to the number of arguments (separated by <s-amp>) in a parameter.

The semantical type-domain is not equivalent to the syntactically defined <info-unit>. There are no syntactical definitions of bytes and words. Therefore - if a parameter-value-definition allows both byte- and word-type-info-units it may result in a conflict due to rule 1:

If a parameter-value contains numbers in the domain (.0..255.) the parameter-definition is primarily consulted for acceptance of byte-types, secondarily for wordtypes. I.e. an argument like

...-H'a-d'1024-...

will introduce a semantic error. Instead byte-like and word-like numbers should be separated in different arguments.

2.4.2.5 Rule 5.

2.4.2.5

A parameter is 'specified' when dedicated a value either by position or by name.

A parameter may be respecified an arbitrary number of times in the same block.

With respect to the semantic analyzer, all parameters are unspecified in the beginning of each block.

2.4.2.6 Rule 6.

2.4.2.6

A parameter is 'specified' by name when a value preceeded by this parameters name occurs in the current block. Specified by position means: A value not preceeded by <id><equal> (see 2.3.1.5) is dedicated to the current parameter.

The current parameter is the parameter in sequence i. e. If you have just specified parameter no. x by name (or position) the current parameter is no. x+1.

If the no. of defined parameter has run out an error occurs.

2.4.2.7 Rule 7.

2.4.2.7

An asterix means: Print out currently dedicated value of the parameter specified by position.

The printoutformat is identical to the lcp-mess-format.

2.5 Session- subsession-definition.

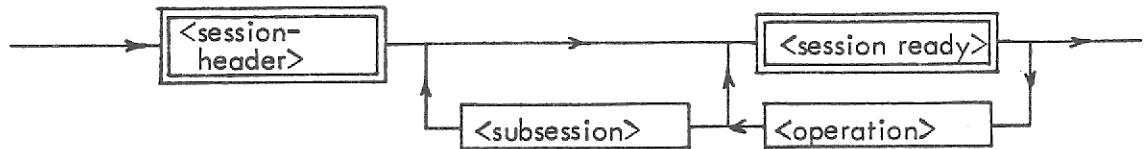
2.5

2.5.1 The syntactical elements.

2.5.1

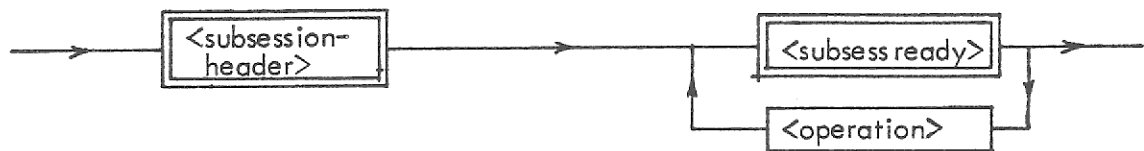
2.5.1.1 <session>

2.5.1.1



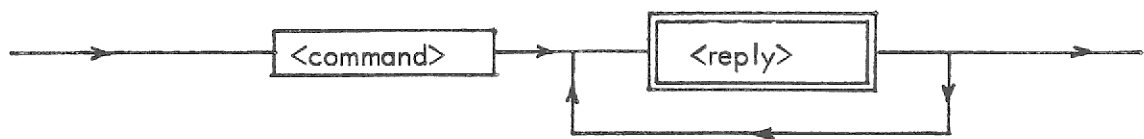
2.5.1.2 <sub-session>

2.5.1.2



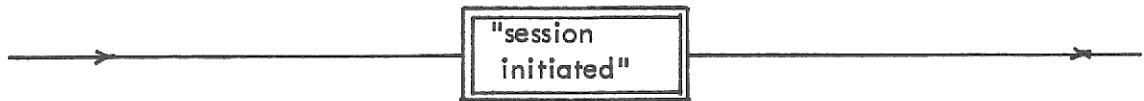
2.5.1.3 <operation>

2.5.1.3



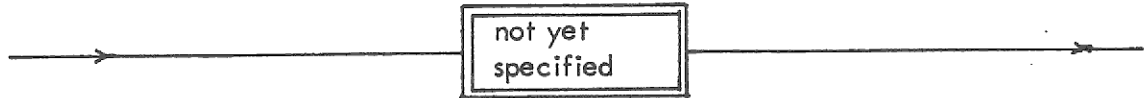
2.5.1.4 <session-head>

2.5.1.4



2.5.1.5 <subsession-head>

2.5.1.5



2.5.1.6 <reply>

2.5.1.6

Command-dependent.

2.5.1.7 <session ready>

2.5.1.7



2.5.1.8 <subsession ready>

2.5.1.8



2.5.2 Operations and subsessions.

2.5.2

A session in NCTH-MML is a sequence of operations on session-scope or a specific scope inner to the session-scope.

An operation is a single command and may be succeeded by a system-reply.

A command may consist of 0 or more blocks. The first block always acts on the current scope (session- or subsession-). The interpretation of the succeeding

blocks depends on the commandcode specified in the prior block.

2.5.2.1 Rule 1.

2.5.2.1

If no operation is specified then the succeeding block acts on current's scope.

If an operation is specified then the succeeding block acts on this specific inner scope.

If an error occurs then all succeeding blocks will be ignored.

2.5.2.2 Rule 2.

2.5.2.2

At scope-entry all parameters will have the value 'nil', i.e. no value dedicated.

When a parameter is dedicated a value (2.4.2.6-7) this value is kept until a new specification in the same or a succeeding command acting on the same scope.

A simple operation always ends with the command, i.e. with the terminator (a semicolon).

Any error detected during command-interpretation will end the operation-execution.

A subsession may be ended with some special command (recommended: "end").

The subsession may consist of several simple operations acting on any outer scope and possibly on some specific inner scope.

The session-scope itself acts as a subsession surrounded by an empty scope. The character "<" will - no matter where it occurs in a command - force an exit to this outer scope. After this exit, the session level will be re-entered immediately, but all parameter-values will be nil due to rule 2.

2.5.3 The scope-concept.

2.5.3

A scope consists of

1. A parameter-vector and
2. A set of constants, which may be of any of the semantically defined info-unit-types.

In the session or a subsession-scope, the first parameter is defined as a commandtype, and represents the specification of the operation to be executed.

The succeeding parameters are defined by

1. Parameter-name.
2. Acceptable Info-unit-types.
3. Max. no. of arguments in a specified value.
4. Max. no. of info-units in a single argument.

Example 1.

In 'session' a parameter h1 is defined as

1. Name = 'h1'
2. Info-units of wordtype only.
3. Max. 3 arguments.
4. Max. (i.e. exact) 1 info-unit pr. argument.

therefore

```
h1 = 1 & 2 & 3;
h1 = 7 & 9;
h1 = b'01101101;
```

are legal specifications, but

```
h1 = "abc";
h1 = 2 - 3 & 4;
```

are illegal specifications.

3. Session- subsessions and operation-descr.

3.

3.1 Session

3.1

Presently 'session' consists of a parametervector:
 (oper,h1,h2,h3,h4,d1,d2,d3,d4,defbase)
 and constants:
 (sendm,form)

where oper is defined by:

1. Name = 'oper'.
2. Accepted info-unit-types =
 (. commandtype .)
3. Max. 1 argument.
4. Max. 1 info-unit pr. arg.

h1 - h4 is defined by

1. 'h1' - 'h4'.
2. (. wordtype .)
3. 3.
4. 1.

d1 - d4 is defined by

1. 'd1' - 'd4'.
2. (. bytet., wordt., symb-n-t., textt. .)
3. no limits.
4. no limits.

defbase is defined by

1. 'defbase'.
2. (. bytetype .)
3. 1.
4. 1.

Both constants are of commandtype.

The parameter oper represents the specification of the operation to be executed.

The parameters 2. - 9. are used by the operation sendm as defaultvaules. Therefore consult sec. 3.2 for the description of these parameters.

The 10. parameter 'defbase' represents the basevalue used as default-base by lexan (sec 2.2). Other basevalues than (. 2,8,10,16 .) generates an errormessage.

The constants: sendm and form are of commandtype and used as operation-specifying values.

Example 2. Legal operations on session-scope.

```
dl="test"&l - 2-3& H`a123&l0&&l0;
sendm,576&0&4;
,*,,,*;
defbase = 16 ; note: the old defbase-value = d`10
; this is a comment. ?234&$%&`(`
```

3.2 Sendmess

3.2

The sendmess-scope consists of the parametervector
(headno,datano,reclcp,sptype,lcpoper,chdata)
and no constants.

Headno is defined by

1. 'headno'.
2. (. bytetype .)
3. 1.
4. 1.

Datano is defined by

1. 'datano'.
- 2-4. as for headno.

Reclcp is defined by

1. 'reclcp'.
2. (. wordtype .)
3. 1.
4. 1.

Sptype is defined by

1. 'sptype'.
- 2-4. as for headno.

Lcpoper is defined by

1. 'lcpoper'.
- 2-4. as for headno.

Chdata is defined by

1. 'chdata'.
- 2-4 as for dl-d4 in session.

Sendmess fills out a supervisor-message destined to a specific LCP. Sendmess is an operation called from session by use of operation = sendm.

The headno-parameter specifies which of the parameters hl-h4 from the session-scope is to be used by headfield-specifications. If headno-value is outside (. 1..4 .) the defaultvalue 1 is assumed.

The datano-parameter specifies wich of the parameters dl-d4 is to be used by data-field-specification. As explained in sessiondescription the parameters dl-d4

may contain any number of arguments containing any number of info-units. The datafieldformat is a simple array of bytes, wherefore the bit-layout must be known by the user.

The reclcp-, sptype- and lcpoper-parameters are used in the following headfieldspecifications.

The chdata-parameter is used in the same manner as dl-d4, just that the value of this parameter is concatenated to the value of the parameter selected by datanr.

The databuffers used for messagetransport is delivered by NCP. If no databuffer is pending at NCTH a message is printed:

'no ncp-buf available'

and the sendmess-operation is neglected.

3.2.1 The headfieldspecifications.

3.2.1

The sendmess-operation sets the headfieldvalues as defined in the following using a pseudo-Algol-like language:

```
receiverid :=
  (0, IF reclcp <> I
    THEN reclcp
    ELSE IF h(headno).arg(1) <> I
      THEN h(headno).arg(1)
      ELSE 0 );

senderid := (0,32766);

seqno := counter mod 256;
<* counter is increased after sendmess-operation *>

ord(sptype.reqans) :=
  IF sptype <> I
  THEN sptype.bit(7)
  ELSE IF h(headno).arg(2) <> I
    THEN h(headno).arg(2).bit(7)
    ELSE 0;

ord(sptype.messev) := <* as reqans but using bit(6)
*>;

ord(sptype.ncpcontr) := <* as reqans but using bit(5)
*>;
```

```
ord(sptype.repfunc) := <# as reqans but using bit(4)
*>;
```

```
ord(sptype.rejfunc) := <# as reqans but using bit(3)
*>;
```

```
ord(sptype.statcntr) := <# as reqans but using bit(2)
*>;
```

```
lcpoper :=
  IF lcpoper <> I
  THEN (( lcpoper MOD 256 ) DIV 4, lcpoper MOD 4)
  ELSE IF h(headno).arg(3) <> I
        THEN ((h(headno).arg(3) MOD 256) DIV 4),
              h(headno).arg(3) MOD 4
        ELSE (0,0);
```

```
status := ( 16***0 );
```

```
time := ( timelgt***0);
```

```
bytecount :=
  (IF datano IN (. 1..4 .)
   THEN IF d(datano) <> I
         THEN length( d(datano) )
         ELSE 0
   ELSE 0 ) +
  (IF chdata <> I
   THEN length(chdata)
   ELSE 0 );
```

```
data :=
  (IF datano IN (. 1..4 .)
   THEN d(datano)
   ELSE I ) conc_with
  chdata;
```

where

I means nil.
 h(—) means session-parameter hl-h4
 d(—) — — — — — dl-d4
 h(—).arg(x) means the xth argument
 of hl-h4.
 h(—).arg(x).bit(y) means the yth
 bit of the xth arg of hl-h4.
 length(arg) gives the no. of bytes in
 arg.
 arg1 conc_with arg2 concatenates the

bytestrings of arg1 and arg2.

Example 3. Legal operations on sendmess-scope.

```
sendm : 1,1;  
sendm : reclcp = 576, 0 , 5 ;  
sendm,0&0&5 : reclcp=576;
```

3.3 C-formatter.

3.3

The c-formatter-scope contains the parameter vector:
 (oper,prhead,prdata,prvert,keeplm,mask)
 and constants
 (replm)

where

oper is defined as oper in session.

prhead is defined by

1. 'prhead'.
2. (. bytetype .) .
3. 1.
4. 1.

prdata is defined by

1. 'prdata'.
- 2-4. as prhead.

prvert is defined by

1. 'prvert'.
- 2-4. as prhead.

keeplm is defined by

1. 'keeplm'.
- 2-4. as prhead.

mask is defined by

1. 'mask'.
2. (. bytetype .) .
3. 1.
4. 16.

The c-formatter-operation is accessible from session by use of oper = form. This operation gives access to the printoutoptions described in sec 1.4 via an option-mask of 16 bits:

(maskbit(0) .. maskbit(15)) .

If bit(0) = 0 then the messagehead is suppressed.

If bit(1) = 0 then the datapart is suppressed.

If bit(2) = 0 then the data is printed out in horisontal format.

If bit(3) = 0 then the last message printed is returned immediately to NCP else it is kept and the previous message is returned.

The rest of the maskbits are unused (for future use).

By using the operation replm it is possible to reprint the last message.

3.3.1 The maskbitspecification.

3.3.1

As for sendmess a formal-like specification of the mask-bit-setting follows.

```
maskbit(0) :=
  IF mask(0) <> I
  THEN ( mask(0) > 0 )
  ELSE IF prhead <> I
        THEN ( prhead > 0 )
        ELSE maskbit(0) ;

maskbit(1) := < * as maskbit(0) but using mask(1) and
prdata * >;

maskbit(2) := < * as maskbit(0) but using mask(2) and
prvert * >;

maskbit(3) := < * as maskbit(0) but using mask(3) and
keepm * >;

FOR i := 4 TO 15 DO
  maskbit(i) :=
    IF mask(i) <> I
    THEN ( mask(i) > 0 )
    ELSE maskbit(i) ;
```

where

I means nil.

maskbit(i) means then ith bit of the internal option-mask.

mask(i) means the (i+1)th info-unit of the
6th parameter (mask).

Example 4.

Legal operations on c-formatter-scope :

```
form: ,1,0,1;  
form: keep1m = 0;  
form: prvert= 1 : replm;
```

<u>4. Errormessages.</u>	4.
<u>4.1 Introduction.</u>	4.1
Errors detected by the comint-facilities described in sec. 1.3 are noted to operator as messages of the format described in sec. 1.4.2	
<u>4.2 Errormessage-description.</u>	4.2
<u>4.2.1 Error 0</u>	4.2.1
Internal error.	
<u>4.2.2 Error 1 Missing separator.</u>	4.2.2
Two lexical symbols (as <number> <text>) can not be distinguished due to a missing separator-character.	
<u>4.2.3 Error 2 Timeout.</u>	4.2.3
More than 20 seconds between two keystrokes.	
<u>4.2.4 Error 3 Illegal digit.</u>	4.2.4
A prefix-number contains a digit-value larger than or equal to prefix-base.	
<u>4.2.5 Error 4 Illegal character.</u>	4.2.5
A reserved character (for future use) has been used outside text-strings.	
<u>4.2.6 Error 5 Number overflow.</u>	4.2.6
A prefix-number-value is too big.	
<u>4.2.7 Error 6 Char-buf overflow.</u>	4.2.7
The internal characterbuffer is filled up. If this error is permanent try to reset session by entering the character '<'. The character '<'.	

- 4.2.8 Error 7-10 4.2.8
Internal errors.
- 4.2.9 Error 11 Misplaced '=' 4.2.9
Either too many '=' or missing ', ' or ': ' .
- 4.2.10 Error 12 Missing object. 4.2.10
Two separators succeeds each other (except ', ' and ': ') .
- 4.2.11 Error 13 Missing param-sep. 4.2.11
Possibly two successive '*' .
- 4.2.12 Error 14 Syntax-error detected. 4.2.12
Unintelligible syntactical construction.
- 4.2.13 Error 15-20 4.2.13
Internal errors.
- 4.2.14 Error 21 Byte-word-conflict. 4.2.14
Either neither bytes nor words are allowed by parameter-definition, or both are allowed, and you have mixed them in the same argument. Consult sec. 2.4.2.4 .
- 4.2.15 Error 22 Integer-type illegal. 4.2.15
Neither bytes nor words are allowed by parameter-definition.
- 4.2.16 Error 23 Number-buf overflow. 4.2.16
The internal number-buffer is filled. Reaction as for error nr. 6.

- 4.2.17 Error 24 String-type illegal. 4.2.17
 Strings (text or symb-name) are not allowed by parameter-definition.
- 4.2.18 Error 25 Illegal base value. 4.2.18
 You have specified an illegal base-value.
 Note: use a prefix-number when specifying defbase.
- 4.2.19 Error 26 Command-type illegal. 4.2.19
 Command-type-constants not allowed by this param.-def.
- 4.2.20 Error 27 Unknown Identifier. 4.2.20
 Identifier wrongly spelled or used.
- 4.2.21 Error 28 Too many objects. 4.2.21
 Consult sec. 2.4.2.1 .
- 4.2.22 Error 29 Diff. obj-types illegal. 4.2.22
 Consult sec. 2.4.2.1 .
- 4.2.23 error 30 Illegal subval-arg 2. 4.2.23
 Right-hand-side of sub-val is a number greater than 255 .
- 4.2.24 Error 31 Illegal subval-arg 1. 4.2.24
 Right-hand-side of sub-val is a non-byte type.
 Consult sec. 2.4.2.3 .
- 4.2.25 Error 32 Too many arguments. 4.2.25
 Consult sec. 2.4.2.4 .
- 4.2.26 Error 33 Unknown parameter-id. 4.2.26
 As error 27.

4.2.27 Error 34 Too many param-spec.

4.2.27

An attempt to specify a value (by position) to a non existing parameter.
Consult sec. 2.4.2.6 .

4.2.28 Error 35-50

4.2.28

Internal errors.

4.3 At internal errors.

4.3

Please inform RC (UH) and specify error-nr. and action causing the error.

5. References.

5.

(1) Series Z Recommendations (Z.311 - Z.359) Part II
CCITT.

(2) Study group XI - Contribution No. 269 June 1978.
CCITT COM XI No. 269-E.

(3) NCP DATA STRUCTURES by Claus Houlberg Hansen.
December 1980. Revision 1.01.

RETURN LETTER

Title: NCTH User's Guide

RCSL No.: 43-GL11149

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

..... Fold here

..... Do not tear - Fold here and staple

Affix
postage
here

 **REGNECENTRALEN**
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark