

## NOTE:

02

This document represents the decisions reached by a combined Command Language / Novice-expert review groups meeting held the mornings of 29-May and 30-May.

03

With the NLS Utility will come a substantial increase in our NLS user community. These, in general, are users which we would like to please.

Consequently, I herein propose modifications to the command language which I (chi) feel make it simpler, more consistent and somewhat more complete (although, I think it has a way to go before we can call it complete). I have discussed most of these issues with the "novice user" and "command language" groups. I think these changes should be in the running system BEFORE we begin training these new users (say by mid JUNE)! It is quite difficult to learn a command language which is changing while you are learning it!

04

The following changes should be made in the command language:

05

The command language should be made to consist of an editor and special purpose subsystems. The command language for each should consist of frequently used commands, which are recognized by their first letter (unless preceded by a space) and infrequently used, new, or experimental commands, which are preceded by a SPACE and which are recognized when the user types enough characters. There should be a consistent VERB-NOWN form to commands and verbs should be used in a consistent manner.

06

This allows commands to be reasonably named and added without worrying about first letter conflicts while "protecting" frequently used commands -- significant problems currently.

07

This should apply to operand types also, of course, but may not be widely used at first.

08

In "novice" mode, a system supplied SPACE will precede each command the user gives.

09

Subsystem names should be recognized when the user has typed enough characters for uniqueness. All subsystems should terminate with the "Quit" command, as should NLS. All subsystems should have Execute, Goto, and help commands.

010

The concept of Address Expression should be generalized for DNLS, TNLS, and DEX such that wherever a statement name or number is currently used, an appropriate ADDRESS EXPRESSION should be allowed (see Appendix C).

011

For DNLS, a selection should be defined as

012

SEL = (BUG / OPTION DAE CA);

013

Note that the use of markers in DNLS by holding the rightmost mouse button down and typing the marker name should be eliminated, since one will be able to type OPTION and an arbitrary DAE followed by a CA.

014

In TNLS a selection should be defined as	015
SEL = DAE CA;	016
Note the syntactic conflict inherent in SEL / LIT alternatives in commands. This can be avoided by converting any such choice points in TNLS to LIT / OPTION SEL.	017
In DEX a selection is defined as	018
SEL = STAE CA;	019
Where DAE (Dynamic Address Expression) is defined in Appendix C, and STAE (Static file Address Expression) is defined in the DEX-II design document.	020
A Dynamic Address Expression should be consistent with existing links, the same DAE should work in TNLS and in DNLS, and the elements of the expression should be reasonably mnemonic. A DAE should be available in NLS wherever a statement number, SID, or statement name is now used (as in links, jumps, etc.).	021
The replacement of the statement name/number field by a DAE provides a powerful extension to the link syntax and will be compatible with extant links.	022
Editing command changes	023
Terminating any editing command with INSERT or REPEAT (what used to be called CDOT, for historical reasons) is shorthand for Command Accept (CA) followed by the INSERT or REPEAT commands, respectively.	024
The INSERT command allows one to quickly insert a new statement after the CM.	025
The REPEAT command allows one to repeat the last used editing command, perhaps defaulting one of the operands to the Control Marker (CM) instead of asking the user to select it.	026
The notion of operand-type defaulting in DNLS should be eliminated.	027
Substitute changes	028
For the new form of the substitute command, see Appendix B.	029
An elipses capability should be made available in the substitute command. That is, text...text should be allowed for the specification of text to be replaced. This should result in instances of <text1 arbitrary-text text2> being replaced in the substitute.	030

This is subject to the constraint that the <text1...text2> be in one statement, and that an occurrence of text1 can only be paired with the first occurrence of text2 following text1. 031

If text1 is null then assume String Front (SF); if text2 is null then assume String End (SE). 032

If one wishes to actually substitute for a string of three periods (...), one must precede each of the three periods by Literal Escape (LE). 033

Jump command changes 034

SP command in TNLS will be replaced by Jump to. 035

The Jump commands should be made to be like the rest -- no state of its own. 036

Jump to successor, Jump to Predecessor should require one to type 'J 'S 'J 'P just as Insert Character, Insert Word requires one to type 'I 'C 'I 'W. 037

The sub commands of Jump to End should be deleted. 038

The Jump File Link command should be removed since Jump to Link is equivalent. 039

The order of operand selection in Move, Copy, Append, Assimilate, and Substitute should be changed. Please see the command syntax in Appendix B. Basically, the move/copy/assimilate should be of the form "Move/Copy/Assimilate This to There", rather than its current form "Move/Copy/Assimilate to There, This". Append should be "Append this to that", instead of "Append to that, this". 040

Note that this could be considered as being inconsistent with the insert command, although this inconsistency might prove minor and unimportant. 041

A more important problem is that the first operand selection in a command (given our current approach for switching commands) must allow us to differentiate between a keyword and a LIT. Consequently, we would not be able to allow move/copy/append to have LIT as a possible first operand. 042

A new control character "OPTION" should be available for specifying optional arguments to commands. 043

In addition, I propose that there be a FILTER command for content filters, a RESEQUENCE SID's command, and a COMPACT FILE command (doing away with Output File). Please refer to Appendix B for details. 044

Appendix'A: Proposed Command Language (summary) 045

Commands which must be preceded by SP are preceded by SP in this list:

	append	046
	<del>allow private modifications to file (old browse mode)</del>	048
	<del>assemble program</del>	049
plume ←	<del>assimilate</del>	050
	<del>arm NLSDDT ↑H</del>	051
	break	052
	copy	053
	<del>&lt;copy&gt; file</del>	054
	<del>clear window</del>	055
	<del>compact file</del>	056
	<del>compile program</del>	057
	<del>connect (display / tty)</del>	058
	<del>create file</del>	059
	delete	060
	<del>&lt;delete&gt; markers</del>	061
	<del>&lt;delete&gt; file</del>	062
	<del>disarm NLSDDT ↑H</del>	063
	<del>disconnect terminals</del>	064
	execute SUBSYSTEM-NAME command	065
	edit	066
	expert	067
	expunge directory	068
	goto SUBSYSTEM-NAME	069
	help	070
	insert	071
	<del>&lt;insert&gt; (sequential / assembler) file</del>	072
	<del>&lt;insert&gt; Journal submission form</del>	073

~~jump (DNLS and TNLS)~~  
~~k (unused)~~  
~~load file FILENAME CA;~~  
~~move~~  
~~<move> boundary~~  
~~<move> file~~  
~~mark~~  
~~merge~~  
~~novice~~  
~~output~~  
~~p (unused in DNLS) print (TNLS)~~  
~~playback session~~  
~~quit [SUBSYSTEM-NAME / NLS]~~  
~~replace~~  
~~receive connection from terminal~~  
~~record session~~  
~~relock file~~  
~~resequence SID's in file~~  
~~reset (tty-simulation window /  
viewspecs)~~  
~~substitute~~  
~~set (case /  
character size for window /  
filter /  
link default for file /  
name delimiters /  
tty-simulation window /  
viewspecs)~~  
~~show (file status /  
marker list /  
name delimiters /  
viewspecs [verbose] status)~~  
~~simulate terminal type (display / TI-terminal / etc.)~~

Look { at file/die/success }  
 for ~~control~~ 075 }

{ replaced by x-file editing }

formatted output

076  
 077  
 078  
 079  
 080  
 081  
 082  
 083  
 084  
 085  
 086  
 087  
 088  
 089  
 090  
 091  
 092  
 093  
 094  
 095  
 096

sort/ <del>wrap</del> ?	097
<del>split window (vertically / horizontally) (DNLS)</del>	098
transpose	099
<del>terminate (recording / private modifications to file)</del>	0100
update file	0101
<del>unlock file</del>	0102
<del>verify file</del>	0103
w (not used)	0104
x (unused)	0105
y (unused)	0106
z (unused)	0107
' . (TNLS) show CM	0108
' / (TNLS) type context of CM	0109
' ; Comment	0110
' \ (TNLS) print statement	0111
' ↑ (TNLS) print back statement	0112
linefeed (TNLS) print next statement	0113
<del>INSERT %Insert Statement after Control Marker%</del>	0114
<del>REPEAT last editing command</del>	0115
<del>TAB %to next occurrence of content or word%</del>	0116
<del>?</del>	0117

## Appendix 'B: Proposed Command Language (detail)

### Definitions

CA = ↑D / CR (default);	0120
%SEL or command terminator%	0121
REPEAT = CDOT = ↑B / ESC/ALTMODE (default);	0122
%Terminate current editing command and begin REPEAT command, possibly defaulting a selection to CM%	0123

```

INSERT = ↑E / user-settable characters;                                0124
    %Terminate current editing command and begin INSERT command%      0125
CD = ↑x / DEL/RUBOUT (default);                                       0126
    %abort current command specification%                              0127
OPTION = ↑U / user-settable characters;                                0128
    %use optional parameter or use optional form of a command%       0129
Confirmation = CA / INSERT / REPEAT;                                   0130
    %used to confirm editing commands%                                 0131
LEVADJ = $('u / 'd) (SP / CA);                                         0132
TextSpec1 =                                                            0133
    'c <... Character> / 'w <... Word> / 'v <... Visible> / 'i
    <... Invisible> / 'l <... Link> / 'n <... Number>;                0134
TextSpec2 =                                                            0135
    't <... Text>;                                                    0136
Structurespec1 =                                                       0137
    's <... Statement> / 'b <... Branch> / 'p <... Plex>;           0138
Structurespec2 =                                                       0139
    'g <... Group>;                                                  0140
Where <... WORD> denotes that WORD is appended to the command
feedback.                                                                0141
FILENAME = LIT / SEL;                                                  0142
    %where lit is of the form <dir>file.ext%                          0143
LIT =                                                                    0144
    literal text typed by the user, excluding control characters
    such as CA, CD, INSERT, OPTION, etc. unless preceded by the
    literal escape (LE) character (default LE is control-V).          0145

```

Command Language (Note: All commands and operand-types may be preceded by a SP and some MUST be. If a command or operand-type is not preceded by a SP, the system default command or operand-type for the first letter typed will be assumed and used. If it is preceded by a SP then recognition will take place when a sufficient number of

characters have been typed to determine uniqueness.) 0146

append 0147

Syntax: 'a <Append> 0148

((TextSpec1 / StructureSpec1) <at> SEL / 0149

(TextSpec2 StructureSpec2) <from> SEL <(to)> SEL) 0150

<to> SEL LIT 0151

Confirmation; 0152

allow private modifications to file (old browse mode) 0153

assemble program 0154

Syntax: " asse" <Assemble Program at> SEL <Using> 0155

ASSEMBLER-NAME <to file> FILENAME CA; 0156

assimilate 0156

Syntax: " assi" <Assimilate> 0157

{StructureSpec1 SEL / StructureSpec2 SEL SEL) 0158

<after Statement> SEL LEVADJ VIEWSPECS 0159

Confirmation; 0160

arm NLSDDT ↑H 0161

break 0162

Syntax: 'b <Break> 0163

((TextSpec1 / TextSpec2) <at> SEL / 0164

(StructureSpec1 / StructureSpec2) <at> SEL LEVADJ LIT ) 0165

Confirmation; 0166

IT would be extremely nice if break Plex ad break group 0167

allowed us to convert a structure like 0168

statement a 0169

statement b 0170

statement c 0171

statement d 0172



into a structure like	0173
	0174
statement a	0175
statement b	0176
new statement	0177
statement c	0178
statement d	0179
by breaking at statement b.	0180
We could define break word, text etc. as just a quick insert of a SP. Break statement and branch would be equivalent (unless we wish to distinguish??).	0181
copy	0182
syntax: 'c <Copy>	0183
((TextSpec1 <at> SEL / TextSpec2 <from> SEL <(to)> SEL) <to follow> SEL /	0184
(StructureSpec1 <at> SEL / StructureSpec2 <from> SEL <(to)> SEL) <to follow> SEL LEVADJ )	0185
Confirmation;	0186
<copy> file	0187
allows users to copy files from NLS.	0188
clear window	0189
compact file	0190
compile program	0191
Syntax: " com" <Compile Program at> SEL <Using> COMPILER-NAME <to file> FILENAME Confirmation;	0192
connect (display / tty)	0193
syntax: 'c <Connect>	0194
('d <Display> / 't <TTY>) <to terminal> NUMBER ['i <Input and output> / 'o <Output only>] CA;	0195
"Input and output" type connection requires that the recipient issue a Receive connection command.	0196

create file	0197
Syntax: " cr" <Create File> FILENAME Confirmation;	0198
Creates a new (empty) file.	0199
delete	0200
Syntax: 'd <Delete>	0201
((TextSpec1 / StructureSpec1) <at> SEL /	0202
(TextSpec2 / StructureSpec2) <from> SEL <(to)> SEL)	0203
Confirmation;	0204
<delete> markers	0205
<delete> ('a <all markers> / 'm <marker named> LIT) CA;	0206
<delete> file	0207
allows users to delete files from NLS (will take care of Partial Copies).	0208
disarm NLSDDT ↑H	0209
disconnect terminals	0210
execute SUBSYSTEM-NAME command	0211
edit	0212
expert	0213
The Novice/Expert design group should specify this command.	0214
expunge directory	0215
goto SUBSYSTEM-NAME	0216
calculator subsystem	0217
identification subsystem	0218
journal subsystem	0219
includes submission, number assignment, secondary distribution, etc.	0220
measurement subsystem	0221
programs subsystem	0222
query subsystem	0223

user options subsystem	0224
The Novice/Expert design group should specify this subsystem.	0225
includes execute viewchange and show ...	0226
print parameters, feedback parameters, control characters	0227
<Show>	0228
('s <... Selections> / 'c <... Control Mark> / 'u <... Upper Case> / 'i <... Input Prompts> / 'l <... LEVADJ numbers>)	0229
Confirmation;	0230
help	0231
prints out instructions for new or confused users including definitions of terms used in syntax rules for commands.	0232
insert	0233
Syntax: 'i <Insert>	0234
((TextSpec1 / TextSpec2) <after> SEL /	0235
(StructureSpec2 / StructureSpec1) <after> SEL LEVADJ)	0236
LIT	0237
Confirmation;	0238
<insert> (sequential / assembler) file	0239
<insert> Journal submission form	0240
jump (DNLS and TNLS)	0241
Syntax for DNLS: 'j <Jump to>	0242
( SEL DAE CA VIEWSPECS) /	0243
(	0244
(	0245
'i <... Item> /	0246
's <... Successor> /	0247
'p <... Predecessor> /	0248

'u <... Up> /	0249
'd <... Down> /	0250
'h <... Head> /	0251
't <... Tail> /	0252
'e <... End of Branch> /	0253
'b <... Back> /	0254
'o <... Origin> /	0255
SP "ne" <... Next> /	0256
)	0257
SEL VEIWSPECS	0258
) /	0259
(	0260
'l <... Link> ['l <... Locked>] ( SP LIT / SEL VEIWSPECS) /	0261
%locked is a privileged facility and is not available to the average user%	0262
'r <... Return> CA	0263
%text from 'return' statement%	0264
\$(NOT-CA %text from 'return' statement%) /	0265
'a <... Ahead> CA	0266
%text from 'ahead' statement%	0267
\$(NOT-CA %text 'ahead' from statement%) /	0268
'f <... File>	0269
((SP FILENAME / SEL) VIEWSPECS %load file%/	0270
((('a <... Ahead> /	0271
'r <... Return>)	0272
%file name% \$(NOT-CA %next file name%) /	0273
'n <... Name>	0274
['f <... First> /	0275

'n <... Next>]	0276
(SP LIT CA / SEL) VIEWSPECS /	0277
'c <... Content First>	0278
['f <... First> /	0279
'n <... Next>]	0280
(SP LIT CA / SEL SEL / OPTION %Accept old content%) VIEWSPECS /	0281
'w <... Word First>	0282
['f <... First> /	0283
'n <... Next>]	0284
(SP LIT CA / SEL / OPTION %Accept old word%) VIEWSPECS	0285
)	0286
CA;	0287
Syntax for TNLS: 'j <Jump to> SEL;	0288
k (unused)	0289
load file FILENAME CA;	0290
move	0291
Syntax: 'm <Move>	0292
((TextSpec1 <at> SEL / TextSpec2 <from> SEL <(to)> SEL) <to follow> SEL /	0293
(StructureSpec1 <at> SEL / StructureSpec2 <from> SEL <(to)> SEL) <to follow> SEL LEVADJ)	0294
Confirmation;	0295
<move> boundary	0296
<move> file	0297
allows users to move files from one directory to another from NLS.	0298
mark	0299
Syntax: "ma" <Mark> SEL <with marker name> LIT CA;	0300
merge	0301

Syntax: " mer" <Merge> 0302

((('b <... Branch> / 'p <... Plex>) <at> SEL <into> SEL) / 0303

('g <... Group> <from> SEL <(to)> SEL <into> SEL <(to)> 0304  
SEL))

Confirmation; 0305

novice 0306

The Novice/Expert design group should specify the semantics of 0307  
this command.

output 0308

syntax: 'o <Output> 0309

( 0310

    ('q <Quickprint> / 0311

    'j <Journal Mail Quickprint> / 0312

    'p <Printer> (COM, etc.)) 0313

    FILENAME <Copies = 1?> [NUMBER] ) / 0314

( 0315

    's <Sequential File> / 0316

    'a <Assembler File> ) 0317

    FILENAME ) 0318

CA; 0319

p (unused in DNLS) print (TNLS) 0320

Syntax for TNLS print: 0321

'p <Print> 0322

(StructureSpec1 <at> SEL / StructureSpec2 <from> SEL 0323  
<(to)> SEL) VIEWSPECS [OPTION <using filter:> PATTERN/  
CA / CA;

    if no structure is specified, printing will continue 0324  
    until terminated by control o or until the end of the  
    file is reached.

playback session 0325

quit [SUBSYSTEM-NAME / NLS] 0326

Allows one to terminate NLS from within a subsystem. Also allows one to terminate several levels of subsystem with one command. 0327

replace 0328

Syntax: 'r <Replace> 0329

((TextSpec1 / StructureSpec1) <at> SEL <by> (LIT / rsel) / 0330

(TextSpec2 / StructureSpec2) <from> SEL <(to)> SEL <by> (LIT / r2sel) ) 0331

Confirmation; 0332

where 0333

For DNLS: 0334

    rsel = SEL; 0335

    r2sel = SEL <(to)> SEL; 0336

For TNLS: 0337

    rsel = OPTION SEL; 0338

    r2sel = OPTION SEL <(to)> SEL; 0339

receive connection from terminal 0340

record session 0341

relock file 0342

resequence SID's in file 0343

reset (tty-simulation window / viewspecs) 0344

in TNLS, default (and reset) viewspecs will have statement numbers on (m). Should SID's (I) be on also??? 0345

substitute 0346

syntax: 's <Substitute> 0347

(TextSpec1 0348

    <in> (StructureSpec1 SEL/ StructureSpec2 SEL SEL) 0349

    <New> Collect1 <For Old> Collect1 / 0350

```

TextSpec2                                0351
  <in> (StructureSpec1 SEL/ StructureSpec2 SEL SEL) 0352
  <New> Collect2 <For Old> Collect2)          0353
<Finished?> (NO %repeat at <New>% / YES / OPTION <Using
filter:> PATTERN)                          0354
Confirmation;                              0355
Where:                                     0356
  For TNLS:                                0357
    Collect1 = Collect2 = LIT;              0358
  For DNLS:                                0359
    Collect1 = (LIT / SEL);                 0360
    Collect2 = (LIT / SEL SEL);             0361
    %propagates the current awkwardness in specifying
    a NULL LIT%                             0362

set (case /
  character size for window /
  filter /
  link default for file /
  name delimiters /
  tty-simulation window /
  viewspecs)                               0363

Syntax for case:                            0364
  " ca" <Case>                              0365
    ((TextSpec1 / StructureSpec1) <at> SEL / 0366
    (TextSpec2 / StructureSpec2) <from> SEL <(to)> SEL) / 0367
  'm <Mode>)                                0368
  [mtype]                                   0369
Confirmation;                              0370
  mtype = ('i <initial upper> / 'u <upper> / 'l
  <lower>));                                0371

Note: allows temporary mode setting for a single instance
of the command                             0372

Syntax for Filter: 'f <filter> ('t <to> PATTERN / "on" <On> /

```



"of" <Off> 0373

Syntax for name delimiters: 'n <Name Delimiters> 0374

( 0375

    StructureSpec1 <at> SEL / 0376

    StructureSpec2 <from> SEL <(to)> SEL) 0377

    <Left Delimiter> LIT <Right Delimiter> LIT 0378

    ) 0379

Confirmation; 0380

show (file status /  
marker list /  
name delimiters /  
viewspecs [verbose] status) 0381

name delimiters for statement at > SEL Confirmation; 0382

file [lock / size / ownership / return ring] status 0383

simulate terminal type (display / TI-terminal / etc. ) 0384

sort 0385

Syntax: "so" <Sort> 0386

    (('b <... Branch> / 'p <... Plex>)) <at> SEL / 0387

    'g <... Group> <from> SEL <(to)> SEL) 0388

Confirmation; 0389

split window (vertically / horizontally) (DNLS) 0390

    splits window into two equal halves. 0391

transpose 0392

Syntax: 't <Transpose> 0393

    ((Textspec1 / structurespec1) <at> SEL <and> SEL / 0394

    (Textspec2 / StructureSpec2) <from> SEL <(to)> SEL <and>  
from> SEL <(to)> SEL ) 0395

Confirmation; 0396

terminate (recording /  
private modifications to file) 0397

update file 0398

Syntax: 'u <Update File> %default file name% (OPTION <Old version> / [FILENAME]) CA; 0399

unlock file 0400

verify file 0401

w (not used) 0402

x (unused) 0403

y (unused) 0404

z (unused) 0405

' . (TNLS) show CM 0406

' / (TNLS) type context of CM 0407

' ; Comment 0408

' \ (TNLS) print statement 0409

' ↑ (TNLS) print back statement 0410

linefeed (TNLS) print next statement 0411

INSERT %Insert Statement after Control Marker% 0412

Syntax: INSERT 0413

LEVADJ LIT Confirmation; 0414

Insertion and LEVADJ is relative to CM. 0415

REPEAT last editing command 0416

TAB %to next occurrence of content or word% 0417

' ? 0418

prints the names of all the commands available at the first level, with a comment about typing the first letter of any command followed by a '?' to find out about that (set of) command(s). This should also include an explanation of A:, T:, etc. Also the user is advised to use the command Help to find out about definitions. 0419

Appendix 'C: Definition of Dynamic Address Expression 0420

Dynamic Address Expression elements 0421

location number 0422

A statement number is D \$(L / D / '@). 0423

(no preceding period)		0424
name		0425
A statement name is as defined by the name delimiter routine -- currently defined to be L S(L/ D/ '/' '-).		0426
(no preceding period)		0427
System-supplied Statement IDentifiers (SID's)		0428
'O IS D.		0429
(no preceding period)		0430
A sequence of digits and letters PRECEDED IMMEDIATELY BY A PERIOD can contain the following letters, with associated "Jump" meaning. NOTE: default value for <number> is 1.		0431
[number]'s	jump to successor <number> times	0432
[number]'p	jump to predecessor <number> times	0433
[number]'u	jump to up <number> times	0434
[number]'d	jump to down <number> times	0435
[number]'a	jump to ahead <number> times	0436
[number]'r	jump to return <number> times	0437
[number]"fa"	jump to file ahead <number> times	0438
[number]"fr"	jump to file return <number> times	0439
[number]'o	jump to origin	0440
[number]'e	jump to end	0441
[number]'n	jump to next <number> times	0442
[number]'b	jump to back <number> times	0443
[number]'h	jump to head	0444
[number]'t	jump to tail	0445
[number]'l	jump to the <number>th link	0446
[number]'w	jump to next occurrence of word <number> times	0447
[number]'c	jump to next occurrence of content <number> times	0448

a sequence of digits and letters PRECEDED IMMEDIATELY BY A PLUS (SKIP FORWARD) OR MINUS (SKIP BACKWARD) can contain the following letters, with associated meaning. NOTE, default value of <number> is 1.

	0449
/number/'c skip <number> characters	0450
/number/'w skip <number> word	0451
/number/'v skip <number> visible	0452
/number/'i skip <number> invisible	0453
/number/'n skip <number> number(s)	0454
/number/'l skip <number> link(s)	0455
* name jumps to the next statement by that name	0456
'( text ' ) link	0457
text = directory, filename, DAE : Viewspecc	0458
'/ text ' / content search	0459
text excludes ' / unless preceded by the literal escape character	0460
allows ellipses (...) notation	0461
'< text '> word search	0462
text excludes '> unless preceded by the literal escape character	0463
allows ellipses (...) notation	0464
'; text ';' intra-statement content search	0465
text excludes ';' unless preceded by the literal escape character	0466
allows ellipses (...) notation	0467
'i character character search	0468
< beginning of statement	0469
> end of statement	0470
# text marker name, text = L \$(L/D)	0471
/ print context	0472
\ print statement	0473

note that '/' and '\' are part of a DAE. In DNLS this is accomplished via the two line tty-simulation area above the Command Feedback Area.

0474