
DMCL - A command line emulation for the BAE

Dirk Baechle

v1.0

2006-10-19

Table of Contents

1. Introduction	1
2. Features	2
3. Basic usage	2
4. Installation	2
4.1. Unzip the DMCL archive	2
4.2. Compile the ULCs	2
4.3. Activation	3
5. Commands	5
5.1. General	5
5.2. Grouping commands	7
5.3. Input grid commands	8
5.4. Vertex change commands	9
6. Defined SCM commands	10
6.1. Edit menu	10
6.2. View menu	11
6.3. Symbols menu	12
6.4. Connections menu	13
6.5. Graphics menu	14
6.6. Text menu	15
6.7. Settings menu	15
7. Defined GED commands	15
7.1. Edit menu	16
7.2. View menu	17
7.3. Symbols menu	18
7.4. Routes menu	19
7.5. Areas menu	21
7.6. Text/Drills menu	22
7.7. Settings menu	23

1. Introduction

DMCL is an abbreviation and stands for *Defined Macro Command Line*. It is meant to be an extension to the current BAE keyboard input system, where the user can redefine single keypresses such that a special ULC function is called.

DMCL tries to provide fast access to the most used functions of the SCM and GED module for all people that like to type in commands, instead of clicking their way through menus with the mouse.

It is based on short command words with 2-4 letters, so called mnemonics. They build an orthogonal set of commands, consisting of a *prefix* and *suffix*. The *prefix* equals the action that should be executed, while the *suffix* specifies the object that is about to be changed. For example, in order to add another part/symbol you type the command “as” (Add Symbol). If you want to delete the same object again, simply replace the “a” for “Add” with a “d” meaning “Delete”. So the correct command would be “ds” (Delete Symbol) in the latter case.

This way you are able to access the most often used functions like add/move/copy in a consistent way for all objects that can be created or moved around. Just check out the list of general prefixes and suffixes in Section 5, “Commands” and pick out some easy to remember functions for a start. It's fun and you have functions like “cs” (Copy Symbol) at your fingertips, that are deeply buried in the standard menus otherwise.

2. Features

- Redefines the keys **a-z** (only lowercase), the **RETURN**, **SPACE**, **BACKSPACE**, and **#** key
- Offers a (more or less) orthogonal set of mnemonics for the SCM and GED module
- Special routines for rotating and mirroring symbols/texts/graphics and setting the input grid size

3. Basic usage

- After installation and activation of the DMCL package (see Section 4, “Installation”), you can directly type in your commands (refer to Section 6, “Defined SCM commands” and Section 7, “Defined GED commands” for complete lists) with small letters **a-z**.
- The mnemonic so far is displayed in the status line at the bottom of the BAE window.
- On pressing the **RETURN** key, the current command is executed if it is defined.
- You can repeat the last DMCL command by pressing the **SPACE** bar, the **#** key starts a new mnemonic.
- With **BACKSPACE** you can delete the single last character of your current input (Note that there is no support for scrolling around in the command line by cursor keys.).

4. Installation

4.1. Unzip the DMCL archive

Copy the archive `dmcl.zip` to your BAE installation directory and extract its contents. Under Linux you have to type:

```
> unzip dmcl.zip
```

Windows/DOS users can use `pkunzip` or `WinZip` for uncompressing the package. This creates an additional directory `dmcl` right beside your other BAE folders like `bin` or `baeulc`. It contains all the ULC files for the DMCL extension and patched versions of the `bae.ini` and `uifsetup.ulc`, based on the BAE 6.6 release.

4.2. Compile the ULCs

Change into the `dmlc` directory and compile all the ULCs, except the `uifsetup.ulc`. For Linux users, a small shell script `compall.sh` is provided. Under Windows/DOS you can use the batch file `compall.bat`. Both assume that the DMCL archive was unpacked as described above, else you have to edit the path to the ULC compiler at the top of each script.

Calling

```
> ./compall.sh
```

will start the compilation of all basic routines.

4.3. Activation

Once you successfully compiled all the `dmcl` files/subroutines, the DMCL extension can be activated in two ways: *on request* or *permanent*.

4.3.1. On request

If you like to try out DMCL first, all you have to do is call the single setup routine `dmclinit`. Either type it in at the “User language program name” prompt (File/User Script) or link a special key to this function via the Key programming menu entry (Utilities).

4.3.2. Permanent

If you want to install DMCL such that it is activated automatically on the start of the BAE modules SCM and GED, a few changes have to be applied to your `bae.ini` and `uifsetup.ulc`.

In case you did not add any new settings or entries to these files after the installation of BAE 6.6, you simply call the prepared install script under Linux

```
> ./install_dmcl.sh
```

or

```
> install_dmcl.bat
```

under Windows/DOS.

They perform the two required steps automatically:

1. Copy the file `bae.ini` to the bin directory.
2. Compile the provided `uifsetup.ulc` by calling the ULC compiler.

Else, the following two sections describe the changes that have to be applied if you want to merge the supplied DMCL versions of `bae.ini` and `uifsetup.ulc` with your own modified files. Under Linux it might be helpful to use the `diff` command for comparison. Then you can directly continue with compiling the `uifsetup.ulc` as described in Section 4.3.2.3, “Compiling `uifsetup.ulc`”.

4.3.2.1. Changes to the `bae.ini` file

Add the lines

```
/* ----- dmclinit.ulc ----- */
DMCL_INIT_STD = 1 /* Automatically activates the DM command line
                    for all BAE modules:
                        0 : Not active ->, normal BAE key behav
                        1 : Activate -> all lower case chars, t
                            the Space, Backspace and Return key
DMCL_COMMAND = "" /* Current DM command line */
DMCL_LCOMMAND = "" /* Last DM command line */
DMCL_ELEMSG = "" /* Current message for picking an element */
DMCL_ELTYPE = 0 /* Current element type for picking */
DMCL_C_UNKNOWN DE = 'Unbekanntes DM Kommando!'
DMCL_C_UNKNOWN EN = 'Unknown DM command!'
```

```
DMCL_W_CONTEXT DE = 'Falscher Kontext/Modul für dieses DM Kommando!'  
DMCL_W_CONTEXT EN = 'Wrong context/module for this DM command!'
```

for example right before the entries for `bae_load.ulc`.

By changing the value of `DMCL_INIT_STD` from 0 to 1 and back you can switch the DMCL *on* and *off*.

4.3.2.2. Changes to the `uifsetup.ulc` file

Search for the `TOOLBAR` entry (around l.256) and add the variable `DMCLINIT` to the list of global variables (see l. 5):

```
// Globals  
#define GV_INIREAD "uif_inidone" // bae.ini processed variable  
#define GV_MMBENTRIES "uif_mmbentries" // MMB menu entry count  
int INIDONE /* bae.ini processed flag */;  
int TOOLBAR /* Toolbar activation flag */;  
int DMCLINIT /* DM command line activation flag */;  
int MS_STDMODE /* Mouse mode context function */;  
int MS_STDMODES /* Mouse mode shift context func. */;  
int WINMENU /* Windows menu redefinition flag */;
```

Then search again for `TOOLBAR` (around l.720) and add the line to read the `DMCLINIT` value (see l. 5) as follows:

```
parseinifile();  
varset(GV_INIREAD,1);  
// Set global parameters  
TOOLBAR=bae_iniintval(PAR_TOOLBAR,1);  
DMCLINIT=bae_iniintval("DMCL_INIT_STD",0);  
MS_STDMODE=bae_iniintval(PAR_MSMODE,MS_CONT);  
MS_STDMODES=bae_iniintval(PAR_MSMODES,MS_MOVE);  
WINMENU=bae_iniintval(PAR_WINMENU,1);  
// Set max. dialog box width
```

Search for the string “switch (callertype)” (around l.739, at the end of the main routine) and add the calls to the user program `dmclinit` (see l. 4 and l. 8) which will set up the DM command line environment for the `ULIPSCM` and `ULIPGED` branch of the switch statement:

```
// Define UL interpreter specific keys  
switch (callertype) {  
  case ULIPSCM : setup_scm();  
                 if (DMCLINIT)  
                   runulprog("dmclinit");  
                 break; // Schematic Editor  
  case ULIPGED : setup_ged();  
                 if (DMCLINIT)  
                   runulprog("dmclinit");  
                 break; // Layout Editor  
  case ULIPAR  : setup_ar(); break; // Autorouter  
  case ULIPCAM : setup_cam(); break; // CAM Processor  
  case ULIPCV  : setup_cv(); break; // CAM View  
  case ULIPCED : setup_ced(); break; // Chip Editor  
}  
// Set mouse operation mode
```

4.3.2.3. Compiling uifsetup.ulc

Finally, you have to compile the new `uifsetup.ulc`. Change into the directory with the current file, which is `dmcl` if you use the version provided by this package, and call the `ulc` compiler:

```
> ../bin/ulc uifsetup.ulc
```

Now DMCL is activated by default on the start of the BAE, depending on the state of the flag `DMCL_INIT_STD` (see Section 4.3.2.1, “Changes to the `bae.ini` file”).

5. Commands

This section lists the prefix and suffix tables that show how the mnemonics are constructed, in general and for special modes like *Group commands* or *Input grid* settings. Please, note that not all prefix/suffix combinations are possible!

5.1. General

5.1.1. Prefixes

Prefix	Meaning
a	Add or Adapt
ap	Auto Place (aps = Auto Place Symbols)
b	Browse
c	Copy
d	Delete
db	open/browse DataBase
e	Edit
f	Find
g	Group commands
h	Highlight
i	Input grid commands
k	Keep (<-> Open)
l	Layer assign
m	Move
n	(re)Name
o	Open (<-> Keep)
p	Paste or Place next
q	Query
r	Rotate (add relative rotation)
rl	Rotate Left (add rotation 90 left)
rr	Rotate Right (add rotation 90 right)
rh	Rotate Half (add rotation 180)
rz	Rotation Zero (reset rotation to 0)
ra	Rotation Absolute (set absolute rotation angle)
s	(re)Size
t	Tap (tap bus), Tear (tear wire) or Toggle (toggle highlight)

Prefix	Meaning
u	Undo
v	Verify or Vertex change commands
w	Width change
x	X-flip (mirror on Y axis) or eXchange
y	Y-flip (mirror on X axis)
yt	Toggle mirror On/Off
ym,yo	Mirror On
yn,yz	Mirror Off
z	highlight with Zoom

Still unassigned are: j = ?.

5.1.2. Suffixes

Suffix	Meaning
a	(copper) Area
an	Area with Net (= potential)
ax	Area eXclusion
axa	Area eXclusion Auto
b	Bus
c	Circle
cd	Circle Dotted
cf	Circle Filled
cx	Circle eXclusion area
ct	Circle Temporary exclusion area
d	Drill hole or Distance
e	Entry (= attribute)
f	Frame (= workspace)
g	Graphics
ge	Graphics Edge (= segment)
gs	Graphics Segment (= edge)
gv	Graphics Vertex
h	Highlight or arrow Head
hd	arrow Head Dotted
hf	arrow Head Filled
i	pIn/Gate
j	Junction (= label)
l	Layer
m	Marker
mp	Module Port
n	Net
nl	Net Length
na	Net All

Suffix	Meaning
nal	Net All Length
o	board Outline
p	Polygon
pd	Polygon with Dotted lines
pf	Polygon Filled
q	Quadrangle (= rectangle)
qd	Quadrangle with Dotted lines
qf	Quadrangle Filled
qx	Quadrangle eXclusion area
qt	Quadrangle Temporary exclusion area
r	Route
re	Route Edge (= segment)
rs	Route Segment (= edge)
rv	Route Vertex
rl	Route Length
s	Symbol (e.g. lm1881)
sl	Symbol Logic
sn	Symbol Name (e.g. N1001)
sp	Symbol Pattern
t	Text
tb	Text Block
v	Vertex
w	Wire
we	Wire Edge (= segment)
ws	Wire Segment (= edge)
wv	Wire Vertex
x	eXclusion area (-> ax)
xa	eXclusion area Auto (-> axa)
z	Zero (= origin)

Still unassigned are k = ?, u = ?, y = ?.

5.2. Grouping commands

The following prefixes and suffixes can be appended to the *Group* command “g”.

5.2.1. Prefixes

Prefix	Meaning
p	select Polygon area
x	deselect (eXclude) polygon area
i	Invert (toggle) selection for polygon area
s	Select single elements

Prefix	Meaning
d	Deselect single elements
t	Toggle selection for single elements
a	select All
z	select Zero (deselect all)
r	Reset (clear) selection
m	Move
l	Layer
fl	File Load
fw	File Write
c	Copy
n	Name
h	change Height of
w	change Width of
e	Edit
k	Keep
o	Open

5.2.2. Suffixes

Suffix	Meaning
a	All
c	Collection (a selection with a name assigned)
e	Entry (Attribute)
g	Graphic
j	Junction (Label)
m	Macro
n	Net
nv	Net Vias
p	Polygon
r	Route
rv	Route with Vias
s	Symbol
t	Text
v	Via
w	Wire

5.3. Input grid commands

The following prefixes and suffixes can be appended to the *Input grid* command “i”.

5.3.1. Prefixes

Prefix	Meaning
k	Keep (=lock)

Prefix	Meaning
o	Open (=unlock)
z	Zero (=unlock)
vf	Very Fine grid (10mil)
f	Fine grid (25mil)
s	Small grid (50mil)
n	Normal grid (100mil)
l	Large grid (200mil)
vl	Very Large grid (400mil)
h	Huge grid (800mil)
vh	Very Huge grid (1600mil)
vfm	Very Fine Mm grid (0.25mm)
fm	Fine Mm grid (0.5mm)
sm	Small Mm grid (1mm)
nm	Normal Mm grid (2.5mm)
lm	Large Mm grid (5mm)
vlm	Very Large Mm grid (10mm)
hm	Huge Mm grid (25mm)
vhm	Very Huge Mm grid (50mm)

5.3.2. Suffixes

Suffix	Meaning
a	Angle
g	Grid

5.4. Vertex change commands

The following prefixes and suffixes can be appended to the *Vertex Change* command “v”.

5.4.1. Prefixes

Prefix	Meaning
a	select All vertices
d	Deselect single vertex (remove/delete)
s	Select single vertex (add)
z	Zero vertices (deselect all vertices)

5.4.2. Suffixes

Suffix	Meaning
a	Arc
c	Chamfer
pa	Polygon Arc
pc	Polygon Chamfer

6. Defined SCM commands

This section lists all mnemonics defined for the SCM module (schematic editor), sorted by the order in which they appear in the standard menus.

6.1. Edit menu

Mnemonic	Command	ulsystem Call
u	Undo	#9003
gr	Group Reset	#700
gp	Group Polygon (Select)	#701:s4:s0
gps	Group Polygon (Select) Symbols	#701:s0:s0
gpw, gpr	Group Polygon (Select) Wires	#701:s1:s0
gpg, gpp	Group Polygon (Select) Graphics	#701:s2:s0
gpt	Group Polygon (Select) Texts	#701:s3:s0
gpa	Group Polygon (Select) All	#701:s4:s0
gx	Group Xclude (Polygon) = Deselect	#701:s4:s1
gxs	Group Xclude (Polygon) Symbols	#701:s0:s1
gxw, gxr	Group Xclude (Polygon) Wires	#701:s1:s1
gxx, gxp	Group Xclude (Polygon) Graphics	#701:s2:s1
gxt	Group Xclude (Polygon) Texts	#701:s3:s1
gxa	Group Xclude (Polygon) All	#701:s4:s1
gi	Group Invert (Polygon) = Toggle Selection	#701:s4:s2
gis	Group Invert (Polygon) Symbols	#701:s0:s2
giw, gir	Group Invert (Polygon) Wires	#701:s1:s2
gig, gip	Group Invert (Polygon) Graphics	#701:s2:s2
git	Group Invert (Polygon) Texts	#701:s3:s2
gia	Group Invert (Polygon) All	#701:s4:s2
gss	Group Select Symbols	#702:s0:s0
gsw, gsr	Group Select Wires	#702:s1:s0
gsg, gsp	Group Select Graphics	#702:s2:s0
gst	Group Select Texts	#702:s3:s0
gsm	Group Select Macros	scmgroup:s0:s5
gse	Group Select Entries	scmgroup:s0:s8
gsc	Group Select Collection	scmgroup:s0:s7
gds	Group Deselect Symbols	#702:s0:s1
gdw, gdr	Group Deselect Wires	#702:s1:s1
gdg, gdp	Group Deselect Graphics	#702:s2:s1
gdt	Group Deselect Texts	#702:s3:s1
gdm	Group Deselect Macros	scmgroup:s1:s5
gde	Group Deselect Entries	scmgroup:s1:s8
gdc	Group Deselect Collection	scmgroup:s1:s7

Mnemonic	Command	ulsystem Call
gts	Group Toggle (Selection) Symbols	#702:s0:s2
gtw, gtr	Group Toggle (Selection) Wires	#702:s1:s2
gtg, gtp	Group Toggle (Selection) Graphics	#702:s2:s2
gtt	Group Toggle (Selection) Texts	#702:s3:s2
ga	Group All	scmgroup:s0:s0
gas	Group All Symbols	scmgroup:s0:s1
gaw, gar	Group All Wires	scmgroup:s0:s2
gag, gap	Group All Graphics	scmgroup:s0:s3
gat	Group All Texts	scmgroup:s0:s4
gaa	Group All All	scmgroup:s0:s0
gz	Group Zero	#700
gzs, grs	Group Zero Symbols	scmgroup:s1:s1
gzw, gzs, grw, grr	Group Zero Wires	scmgroup:s1:s2
gzg, gzp, grg, grp	Group Zero Graphics	scmgroup:s1:s3
gzt, grt	Group Zero Texts	scmgroup:s1:s4
gza, gra	Group Zero All	#700
gm	Group Move	#703
gc	Group Copy	#704
gd	Group Delete	#705
gfl	Group File Load	#706
gfw	Group File Write	#707
gnm	Group Name Macro	#708
ght	Group Height Text	scmgroup:s4
gee	Group Edit Entry	scmgroup:s5
gse	Group Select Entry	scmgroup:s6
gnj	Group Name Junction/Label	scmgroup:s7
gnc	Group Name Collection	scmgroup:s10

6.2. View menu

Mnemonic	Command	ulsystem Call
d	Redraw	#100
za	Zoom All	#101
zw	Zoom Window	#104
zp	Zoom Pan	#105
fs	Find Symbol	findsprt:s0
fsn	Find Symbol Name	findsprt:s1
fsl	Find in Symbol List	findsprt:s2
fe	Find Entry/Attribute	findsprt:s3
fj	Find Junction/Label	findsprt:s4
q	Query element	#117
hn	Highlight Net	#110

Mnemonic	Command	ulsystem Call
dh	Delete Highlight	#111
th	Toggle Highlight	#112
ik	Input grid Keep Grid/Angle	bae_set*lock
ikg	Input grid Keep Grid	bae_setgridlock
ika	Input grid Keep Angle	bae_setanglelock
io, iz	Input grid Open Grid/Angle	bae_set*lock
iog, izg	Input grid Open Grid	bae_setgridlock
ioa, iza	Input grid Open Angle	bae_setanglelock
ivf	Input grid Very Fine (10mil)	dmclgrid
if	Input grid Fine (25mil)	dmclgrid
is	Input grid Small (50mil)	dmclgrid
i, in	Input grid Normal (100mil)	dmclgrid
il	Input grid Large (200mil)	dmclgrid
ivl	Input grid Very Large (400mil)	dmclgrid
ih	Input grid Huge (800mil)	dmclgrid
ivh	Input grid Very Huge (1600mil)	dmclgrid
ivfm	Input grid Very Fine Mm (0.25mm)	dmclgrid
ifm	Input grid Fine Mm (0.5mm)	dmclgrid
ism	Input grid Small Mm (1mm)	dmclgrid
im, inm	Input grid Normal Mm (2.5mm)	dmclgrid
ilm	Input grid Large Mm (5mm)	dmclgrid
ivlm	Input grid Very Large Mm (10mm)	dmclgrid
ihm	Input grid Huge Mm (25mm)	dmclgrid
ivhm	Input grid Very Huge Mm (50mm)	dmclgrid

6.3. Symbols menu

Mnemonic	Command	ulsystem Call
as	Add Symbol	#300
aj	Add Junction/Label	#301
amp	Add Module Port	#302
ps, pj	Paste Symbol, Junction	snextsym
ms, mj	Move Symbol, Junction	#304
ds, dj	Delete Symbol, Junction	#305
esn	Edit Symbol Name	#306
msn	Move Symbol Name	#605
me	Move Entry/Attribute	#607
ee	Edit Entry/Attribute	attrset
esp	Edit Symbol Pattern	ssympatt
qsl	Query Symbol Logic	#310
esl	Edit Symbol Logic	logledit
ast	Add/Rewire Symbol Tag	#311

Mnemonic	Command	ulsystem Call
est	Edit Symbol Tag (= Change Tag Mode)	#312
db	Data Base for Symbols	symsel
yms, yos	Y-flip On Symbol	dmclyon
yms, yzs	Y-flip off (Zero) Symbol	dmclyzero
yts	Y-flip Toggle Symbol	dmclytoggle
ys	Y-flip Symbol	dmclyflip
xs	X-flip Symbol	dmclxflip
rs	Rotate (Relative) Symbol	dmclrrel
ras	Rotate Absolute Symbol	dmclrabs
rls	Rotate Left Symbol	dmclrleft
rrs	Rotate Right Symbol	dmclrright
rhs	Rotate Half Symbol	dmclrhalf
rzs	Rotate Zero Symbol	dmclrzero
ns	Name Symbol	scmpart:s2
es	Edit Symbol	scmpart:s4
qs, qj	Query Symbol or Junction	scmpart:s5
fs	Find Symbol	findsprt:s0
fsn	Find Symbol Name	findsprt:s1
fsl	Find in Symbol List	findsprt:s2
fe	Find Entry/Attribute	findsprt:s3
bs	Browse Symbols	scmpart:s8
vj	Verify Junctions/Labels	scmpart:s11
cs	Copy Symbol	scmpart:s14

6.4. Connections menu

Mnemonic	Command	ulsystem Call
aw, ar	Add Wire	#400
ab	Add Bus	#401
tb	Tap Bus	#402
mb	Move Bus	#403
db	Delete Bus	#404
nb	Name Bus	#405
mwv, mrv	Move Wire Vertex	#412
mwe, mre, mws, mrs	Move Wire Edge	#406
tw, tr, twe, tre, tws, trs	Tear Wire Edge	#407
dwe, dre, dws, drs	Delete Wire Edge	#408
dw, dr	Delete Wire	#409
dn	Delete Net	#410
awa	Add Wire Auto	#411
qb	Query Bus	scmcon:s1
hn	Highlight Nets	scmcon:s2:s1

Mnemonic	Command	ulsystem Call
hna	Highlight Nets All	scmcon:s2:s0
zn	Zoom/Highlight Nets	scmcon:s2:s3
zna	Zoom/Highlight Nets All	scmcon:s2:s2
fn	Find Net	scmcon:s6

6.5. Graphics menu

Mnemonic	Command	ulsystem Call
ap, ag	Add Polygon/Graphic	#500
apd, agd	Add Polygon/Graphic Dotted	#501
apf, agf	Add Polygon/Graphic Filled	#502
aa	Add contact Area	#503
mg, mp	Move Graphic	#504
cg, cp	Copy Graphic	#505
dg	Delete Graphic	#506
agv	Add Graphic Vertex	#507
mgv	Move Graphic Vertex	#508
dgv	Delete Graphic Vertex	#508
mge, mgs	Move Graphic Edge	#509
sp	Size Polygon	scmpoly:s1
ymp, yop, ymg, yog	Y-flip On Polygon/Graphic	dmclyon
ynp, yzp, yng, yzg	Y-flip off (Zero) Polygon/Graphic	dmclyzero
ytp, ytg	Y-flip Toggle Polygon/Graphic	dmclytoggle
yp, yg	Y-flip Polygon/Graphic	dmclyflip
xp, xg	X-flip Polygon/Graphic	dmclxflip
rp, rg	Rotate (Relative) Polygon/Graphic	dmclrrel
rap, rag	Rotate Absolute Polygon/Graphic	dmclrabs
rlp, rlg	Rotate Left Polygon/Graphic	dmclrleft
rrp, rrg	Rotate Right Polygon/Graphic	dmclrright
rhp, rhg	Rotate Half Polygon/Graphic	dmclrhalf
rzp, rzg	Rotate Zero Polygon/Graphic	dmclrzero
dp	Delete Polygon	scmpoly:s3
aq	Add Quadrangle	scmpoly:s4:s0:s0
aqd	Add Quadrangle Dotted	scmpoly:s4:s0:s1
aqf	Add Quadrangle Filled	scmpoly:s4:s0:s2
ac	Add Circle	scmpoly:s4:s1:s0
acd	Add Circle Dotted	scmpoly:s4:s1:s1
acf	Add Circle Filled	scmpoly:s4:s1:s2
ah	Add (Arrow) Head	scmpoly:s4:s2
wp, wg	Width (Change) Polygon/Graphic	scmpoly:s5:s0
gwp, gwg	Group Width (Change) Polygon/Graphic	scmpoly:s5:s2

6.6. Text menu

Mnemonic	Command	ulsystem Call
at	Add Text	tbdvscm: 't2'
mt	Move Text	#601
ct	Copy Text	#602
et	Edit Text	#603
dt	Delete Text	#604
st	Size Text	scmtext:s0
ymt, yot	Y-flip On Text	dmclyon
ynt, yzt	Y-flip off (Zero) Text	dmclyzero
ytt	Y-flip Toggle Text	dmclytoggle
yt	Y-flip Text	dmclyflip
xt	X-flip Text	dmclxflip
rt	Rotate (Relative) Text	dmclrrel
rat	Rotate Absolute Text	dmclrabs
rlt	Rotate Left Text	dmclrleft
rrt	Rotate Right Text	dmclrright
rht	Rotate Half Text	dmclrhalf
rzt	Rotate Zero Text	dmclrzero
atb	Add Text Block	scmtext:s11:s0
mtb	Move Text Block	scmtext:s11:s1
ctb	Copy Text Block	scmtext:s11:s2
etb	Edit Text Block	scmtext:s11:s3
dtb	Delete Text Block	scmtext:s11:s4
stb	Size Text Block	scmtext:s11:s5
gstb	Group Select Text Block	scmtext:s11:s7
gdtb	Group Deselect Text Block	scmtext:s11:s8
wt	Width (Change) Text	scmtext:s12:s1

6.7. Settings menu

Mnemonic	Command	ulsystem Call
ez, mz	Edit/Move Zero (origin)	#802
af, sf	Adapt/Smaller Frame	scmbound:s0
lf	Larger Frame	scmbound:s1
ef	Edit Frame	scmbound:s2

7. Defined GED commands

This section lists all mnemonics defined for the GED module (layout editor), sorted by the order in which they appear in the standard menus.

7.1. Edit menu

Mnemonic	Command	ulsystem Call
u	Undo	#9003
gr	Group Reset	#700
gp	Group Polygon (Select)	#701:s4:s0
gps	Group Polygon (Select) Symbols	#701:s0:s0
gpw, gpr	Group Polygon (Select) Wires	#701:s1:s0
gpp, gpg	Group Polygon (Select) Graphics	#701:s2:s0
gpt	Group Polygon (Select) Texts	#701:s3:s0
gpa	Group Polygon (Select) All	#701:s4:s0
gx	Group Xclude (Polygon) = Deselect	#701:s4:s1
gxs	Group Xclude (Polygon) Symbols	#701:s0:s1
gxw, gxr	Group Xclude (Polygon) Wires	#701:s1:s1
gxa, gxg	Group Xclude (Polygon) Graphics	#701:s2:s1
gxt	Group Xclude (Polygon) Texts	#701:s3:s1
gxa	Group Xclude (Polygon) All	#701:s4:s1
gi	Group Invert (Polygon) = Toggle Selection	#701:s4:s2
gis	Group Invert (Polygon) Symbols	#701:s0:s2
giw, gir	Group Invert (Polygon) Wires	#701:s1:s2
gip, gig	Group Invert (Polygon) Graphics	#701:s2:s2
git	Group Invert (Polygon) Texts	#701:s3:s2
gia	Group Invert (Polygon) All	#701:s4:s2
gss	Group Select Symbols	#702:s0:s0
gsw, gsr	Group Select Wires	#702:s1:s0
gsrv	Group Select Routes with Vias	#702:s4:s0
gsp, gsg	Group Select Graphics	#702:s2:s0
gst	Group Select Texts	#702:s3:s0
gsm	Group Select Macros	gedgroup:s0:s17
gsn	Group Select Net	gedgroup:s0:s13
gsl	Group Select Layer	gedgroup:s0:s11
gsc	Group Select Collection	gedgroup:s0:s19
gds	Group Deselect Symbols	#702:s0:s1
gdw, gdr	Group Deselect Wires	#702:s1:s1
gdrv	Group Deselect Routes with Vias	#702:s4:s1
gdp, gdg	Group Deselect Graphics	#702:s2:s1
gdt	Group Deselect Texts	#702:s3:s1
gdm	Group Deselect Macros	gedgroup:s1:s17
gdn	Group Deselect Net	gedgroup:s1:s13
gdl	Group Deselect Layer	gedgroup:s1:s11
gdc	Group Deselect Collection	gedgroup:s1:s19

Mnemonic	Command	ulsystem Call
gts	Group Toggle (Selection) Symbols	#702:s0:s2
gtw, gtr	Group Toggle (Selection) Wires	#702:s1:s2
gtrv	Group Toggle (Selection) Routes with Vias	#702:s4:s2
gtp, gtg	Group Toggle (Selection) Graphics	#702:s2:s2
gtt	Group Toggle (Selection) Texts	#702:s3:s2
ga	Group All	gedgroup:s0:s0
gas	Group All Symbols	gedgroup:s0:s1
gaw, gar	Group All Wires	gedgroup:s0:s2
gag, gap	Group All Graphics	gedgroup:s0:s3
gat	Group All Texts	gedgroup:s0:s4
gae	Group All Entries	gedgroup:s0:s16
gl, gal	Group All in Layer	gedgroup:s0:s11
gaa	Group All All	gedgroup:s0:s0
gz	Group Zero	#700
gzs, grs	Group Zero Symbols	gedgroup:s1:s1
gzw, gzs, grw, grr	Group Zero Wires	gedgroup:s1:s2
gzg, gzp, grg, grp	Group Zero Graphics	gedgroup:s1:s3
gzt, grt	Group Zero Texts	gedgroup:s1:s4
gze, gre	Group Zero Entries	gedgroup:s1:s16
gzl, grl	Group Zero in Layer	gedgroup:s1:s11
gza, gra	Group Zero All	#700
gm	Group Move	#703
gc	Group Copy	#704
gd	Group Delete	#705
gfl	Group File Load	#706
gfw	Group File Write	#707
gnm	Group Name Macro	#708
gk	Group Keep	#709
go	Group Open	#710
gdl	Group Delete Layer	gedgroup:s4
gel	Group Edit Layer	gedgroup:s4
gwt, gwp, gwg	Group Width Text	gedgroup:s8
ght	Group Height Text	gedgroup:s9
gwr, gww	Group Width Route/Wire	gedgroup:s10:0
gnc	Group Name Collection	gedgroup:s15

7.2. View menu

Mnemonic	Command	ulsystem Call
d	Redraw	#100
za	Zoom All	#101

Mnemonic	Command	ulsystem Call
zw	Zoom Window	#104
zp	Zoom Pan	#105
zo	Zoom Outline	layzmbird
fs	Find Symbol	#111
q	Query element	#112
qd	Query Distance	distance
ik	Input grid Keep Grid/Angle	bae_set*lock
ikg	Input grid Keep Grid	bae_setgridlock
ika	Input grid Keep Angle	bae_setanglelock
io, iz	Input grid Open Grid/Angle	bae_set*lock
iog, izg	Input grid Open Grid	bae_setgridlock
ioa, iza	Input grid Open Angle	bae_setanglelock
ivf	Input grid Very Fine (10mil)	dmclgrid
if	Input grid Fine (25mil)	dmclgrid
is	Input grid Small (50mil)	dmclgrid
i, in	Input grid Normal (100mil)	dmclgrid
il	Input grid Large (200mil)	dmclgrid
ivl	Input grid Very Large (400mil)	dmclgrid
ih	Input grid Huge (800mil)	dmclgrid
ivh	Input grid Very Huge (1600mil)	dmclgrid
ivfm	Input grid Very Fine Mm (0.25mm)	dmclgrid
ifm	Input grid Fine Mm (0.5mm)	dmclgrid
ism	Input grid Small Mm (1mm)	dmclgrid
im, inm	Input grid Normal Mm (2.5mm)	dmclgrid
ilm	Input grid Large Mm (5mm)	dmclgrid
ivlm	Input grid Very Large Mm (10mm)	dmclgrid
ihm	Input grid Huge Mm (25mm)	dmclgrid
ivhm	Input grid Very Huge Mm (50mm)	dmclgrid

7.3. Symbols menu

Mnemonic	Command	ulsystem Call
as	Add Symbol	#300
ms	Move Symbol	#301
ds	Delete Symbol	#302
ps	Place next Symbol	#303
esn	Edit Symbol Name	#305
msn	Move Symbol Name	#605
me	Move Entry/Attribute	#609
mi	Move pIn	#314
xi	eXchange pIn/Gate	#306
ev	Edit Vias	#308

Mnemonic	Command	ulsystem Call
aps	Auto Place Symbols	#5009:s0
yms, yos	Y-flip On Symbol	dmclyon
yms, yos	Y-flip off (Zero) Symbol	dmclyzero
yts	Y-flip Toggle Symbol	dmclytoggle
ys	Y-flip Symbol	dmclyflip
xs	X-flip Symbol	dmclxflip
rs	Rotate (Relative) Symbol	dmclrrel
ras	Rotate Absolute Symbol	dmclrabs
rls	Rotate Left Symbol	dmclrleft
rrs	Rotate Right Symbol	dmclrright
rhs	Rotate Half Symbol	dmclrhalf
rzs	Rotate Zero Symbol	dmclrzero
xsn	eXchange Symbol Names	gedpart:s4:s6
fsn	Find Symbol Name	gedpart:s5:s0
fe	Find Entry/Attribute	gedpart:s5:s1
bs	Browse Symbols	gedpart:s6

7.4. Routes menu

Mnemonic	Command	ulsystem Call
aw, ar	Add Wire	#400
awv, arv	Add Wire Vertex	#401
dwy, drv, mwv, mrv	Move/Delete Wire Vertex	#402
awe, are, aws, ars	Add Wire Edge	#403
mwe, mre, mws, mrs	Move Wire Edge	#404
tw, tr, twe, tre, tws, trs	Tear Wire Edge	#405
dwe, dre, dws, drs	Delete Wire Edge	#406
mw, mr	Move Wire/Route	#415
cw, cr	Copy Wire/Route	#416
dw, dr	Delete Wire	#407
dn	Delete Net	#408
kw, kr	Keep Wire/Route	#409:s0
kn	Keep Net	#409:s1
ow, or	Open Wire/Route	#410:s0
on	Open Net	#410:s1
ww, wr	Width change Wire/Route	#411:s0
wn	Width change Net	#411:s1
hn	Highlight Nets	#412:s0
dh	Delete Highlight	#412:s2
th	Toggle Highlight	#412:s3
qv	Query Vias/Pads (Count)	gedvia:s0
gsnv	Group Select Net Vias	gedvia:s1

DMCL - A command line
emulation for the BAE

Mnemonic	Command	ulsystem Call
gsv	Group Select Vias	gedvia:s2:s0
gdv	Group Deselect Vias	gedvia:s2:s1
gtv	Group Toggle Vias	gedvia:s2:s2
av	Add Via	gedvia:s3
mv	Move Via	gedvia:s4
ev	Edit Via	gedvia:s5:s0
gev	Group Edit Vias	gedvia:s5:s1
dv	Delete Via	gedvia:s6
vv	Verify Via (Delete doubled)	gedvia:s7
kv	Keep Via	gedvia:s8
ov	Open Via	gedvia:s9
hna	Highlight Nets All	conutil:s0:s0
hn	Highlight Nets	conutil:s0:s1
zna	Zoom/Highlight Nets All	conutil:s0:s2
zn	Zoom/Highlight Nets	conutil:s0:s3
qn	Query Net	conutil:s1
qi	Query pIns	conutil:s2
vn, vna	Verify Nets (net list) against layout	conutil:s4
qrl, qwl	Query Route/Wire Length	gedtrace:s0:s0:s0
qnl	Query Net Length	gedtrace:s0:s0:s1
qnal	Query Net All Length	gedtrace:s0:s0:s2
qr	Query Routes	gedtrace:s1
vaa	Vertex change All to Arcs	gedtrace:s4:s0
vsa	Vertex change single Select to Arcs	gedtrace:s4:s1
vac	Vertex change All to Chamfers	gedtrace:s4:s2
vsc	Vertex change single Select to Chamfers	gedtrace:s4:s3
vza	Vertex change Zero to Arcs	gedtrace:s4:s4
vda	Vertex change single Deselect to Arcs	gedtrace:s4:s5
vzc	Vertex change Zero to Chamfers	gedtrace:s4:s6
vdc	Vertex change single Deselect to Chamfers	gedtrace:s4:s7
gvaa	Group Vertex change All to Arcs	gedtrace:s4:s8
gvac	Group Vertex change All to Chamfers	gedtrace:s4:s9
gvza	Group Vertex change Zero to Arcs	gedtrace:s4:s10
gvzc	Group Vertex change Zero to Chamfers	gedtrace:s4:s11
tr, tw	Tear Route/Wire	gedtrace:s6
wre, wrs, wwe, wws	Width change Route/Wire Edge	gedtrace:s9
gatd, gstd	Group Select Tear Drops	gedtrace:s10:s0
dtd	Delete Tear Drop	gedtrace:s10:s1
atd	Add Tear Drop	gedtrace:s10:s2
vl	Verify Layers	gedtrace:s12:s0

Mnemonic	Command	ulsystem Call
dl	Delete Layers	gedtrace:s12:s1

7.5. Areas menu

Mnemonic	Command	ulsystem Call
aa	Add Area	#500
aan	Add Area with Net (Potential)	#501
ax, aax	Add Xclusion area	#502
ao, eo	Add board Outline	#503
ap, ag	Add Polygon/Graphic	#504
apf, agf	Add Polygon/Graphic Filled	#505
ma, mg, mp	Move Area/Graphic	#507
ca, cg, cp	Copy Area/Graphic	#508
sa	Size Area	#509
da, dg, dp	Delete Area	#510
aav, agv, apv	Add Area/Graphic Vertex	#511
mav, mgv, mpv	Move Area/Graphic Vertex	#512
dav, dg, dpv	Delete Area/Graphic Vertex	#512
mae, mas, mge, mgs, mpe, mps	Move Area/Graphic Edge	#513
la, lg, lp	Layer Area/Graphics	gedpoly:s0
wa, wg, wp	Width change Area/Graphics	gedpoly:s2:s0
sg, sp	Size Graphics	gedpoly:s3
yma, yoa, ymg, yog, ymp, yop	Y-flip On Area/Graphic	dmclyon
yna, yza, yng, yzg, ynp, yzp	Y-flip off (Zero) Area/Graphic	dmclyzero
yta, ytg, ytp	Y-flip Toggle Area/Graphic	dmclytoggle
ya, yg, yp	Y-flip Area/Graphic	dmclyflip
xa, xg, xp	X-flip Area/Graphic	dmclxflip
ra, rg, rp	Rotate (Relative) Area/Graphic	dmclrrel
raa, rag, rap	Rotate Absolute Area/Graphic	dmclrabs
rla, rlg, rlp	Rotate Left Area/Graphic	dmclrleft
rra, rrg, rrp	Rotate Right Area/Graphic	dmclrright
rha, rhg, rhp	Rotate Half Area/Graphic	dmclrhalf
rza, rzg, rzp	Rotate Zero Area/Graphic	dmclrzero
vapa	Vertex change All to Polygon Arcs	gedpoly:s5:s0
vspa	Vertex change single Select to Polygon Arcs	gedpoly:s5:s1
vapc	Vertex change All to Polygon Chamfers	gedpoly:s5:s2
vspc	Vertex change single Select to Polygon Chamfers	gedpoly:s5:s3
vzpa	Vertex change Zero to Polygon Arcs	gedpoly:s5:s4

Mnemonic	Command	ulsystem Call
vdpa	Vertex change single Deselect to Polygon Arcs	gedpoly:s5:s5
vzpc	Vertex change Zero to Polygon Chamfers	gedpoly:s5:s6
vdpc	Vertex change single Deselect to Polygon Chamfers	gedpoly:s5:s7
gvapa	Group Vertex change All to Polygon Arcs	gedpoly:s5:s8
gvapc	Group Vertex change All to Polygon Chamfers	gedpoly:s5:s9
gvzpa	Group Vertex change Zero to Polygon Arcs	gedpoly:s5:s10
gvzpc	Group Vertex change Zero to Polygon Chamfers	gedpoly:s5:s11
tae, tas, tge, tgs, tpe, tps	Tear Area/Graphic Edge	gedpoly:s6
aqf	Add Quadrangle Filled	gedpoly:s10:s0:s0
aqx	Add Quadrangle Xclusion	gedpoly:s10:s0:s1
aq	Add Quadrangle	gedpoly:s10:s0:s2
aqt	Add Quadrangle Temporary xclusion	gedpoly:s10:s0:s3
acf	Add Circle Filled	gedpoly:s10:s1:s0
acx	Add Circle Xclusion	gedpoly:s10:s1:s1
ac	Add Circle	gedpoly:s10:s1:s2
act	Add Circle Temporary	gedpoly:s10:s1:s3
ahf	Add arrow Head Filled	gedpoly:s10:s2:s0
ah	Add arrow Head	gedpoly:s10:s2:s1
aaxa, axa	Add area Xclusion Auto	gedpoly:s14:s0
dxa, daxa	Delete area Xclusion Auto	gedpoly:s14:s1

7.6. Text/Drills menu

Mnemonic	Command	ulsystem Call
at	Add Text	tbdvlay: 't1023'
mt	Move Text	#601
ct	Copy Text	#602
et	Edit Text	#603
dt	Delete Text	#604
ad	Add Drill	#607
dd	Delete Drill	#608
ed	Edit Drill	#610
st	Size Text	gedtext:s0
lt	Layer assign Text	gedtext:s1
wt	Width (Change) Text	gedtext:s2:s0
ymt, yot	Y-flip On Text	dmclyon
ynt, yzt	Y-flip off (Zero) Text	dmclyzero
ytt	Y-flip Toggle Text	dmclytoggle

Mnemonic	Command	ulsystem Call
yt	Y-flip Text	dmclyflip
xt	X-flip Text	dmclxflip
rt	Rotate (Relative) Text	dmclrrel
rat	Rotate Absolute Text	dmclrabs
rlt	Rotate Left Text	dmclrleft
rrt	Rotate Right Text	dmclrright
rht	Rotate Half Text	dmclrhalf
rzt	Rotate Zero Text	dmclrzero
atb	Add Text Block	gedtext:s12:s0
mtb	Move Text Block	gedtext:s12:s1
ctb	Copy Text Block	gedtext:s12:s2
etb	Edit Text Block	gedtext:s12:s3
dtb	Delete Text Block	gedtext:s12:s4
stb	Size Text Block	gedtext:s12:s5
ltb	Layer assign Text Block	gedtext:s12:s6
gstb	Group Select Text Block	gedtext:s12:s7
gdtb	Group Deselect Text Block	gedtext:s12:s8
wt	Width (Change) Text	gedtext:s12:s1

7.7. Settings menu

Mnemonic	Command	ulsystem Call
ez, mz	Edit/Move Zero (origin)	#801
af, sf	Adapt/Smaller Frame	gedbound:s0
lf	Larger Frame	gedbound:s1
az	Adapt Zero (to input grid)	gedbound:s2