

Claus Feyling

Fra: Claus Feyling
Sendt: søndag 7. august 2022 10.01
Til: Knut Helland
Kopi: Thomas Benjaminsen; Eirik Hovind; Sveinung Nøding
Emne: 2022-08-07 CODING STYLE – LISP DRAWING FUNCTIONS – ASCII vs UNICODE

For å få max læring ut av dette så tar vi et code review på Teams når det passer.
Alle burde ha «grunnkurs i RC-LISP» - så la oss ta «investeringen» i en time med alle?

Teksten jeg skriver nedenfor blir lagt inn på Github,

Jeg har notert ting jeg la merke til nedenfor. Vi kan bruke Github History når vi skal se gjennom dette.
Jeg fikk for øvrig ikke `_uOSLASH_` og `_OSLASH_` til å fungere og måtte kripe til korset og erstatte med «O» hhv «o»
😞

Mvh Claus

2022-08-07 CODING STYLE – LISP DRAWING FUNCTIONS – ASCII vs UNICODE

- Bundle as much as possible the things that vary among administrations, close to the top of your function declaration. It makes translations and overview / code review simpler.
- The 'description' text – below a symbol's graphics - shall not include the `_OCS_NAME_` text constant
- Avoid using «low-level» calls to AutoCAD funksjoner using the «(command ...)» construct. We have devised our own API to sit between our drawing routines and the AutoCAD API.

`(command _LINE_ p1 p2 _ENTER_)` ; Incorrect formulation – you don't have control over the layer name (or other style issues)

`(DrawLine layer_ThisLayerName p1 p2)` ; Correct formulation. The DrawLine function may be modified once for all its uses, when needed (for instance: Trace).

- «The (setq) and its final parenthesis «)» shall occupy one separate row each when declaring numbered points and constants for use inside a graphics drawing routine.
- In general, don't use plus, minus, division etc to produce the numbers needed for your graphics – they will be hard to read and understand to the one who shall take over the code or the writer himself when maintaining his own code. Rather set X and Y axis scales and use 3-decimal constants to multiply with those scales to produce your numbered points. 4 decimals is too much.
Multiplication with relative numbers from -0.500 to +0.500 (or from -1.000 to +1.000) are much easier to 'see' and to maintain:

```
(setq
  myScaleX      5.0
  myScaleY      7.0
  p1            (list (* -0.123 myScaleX) (* 0.475 myScaleY))
  p2            (list (* 0.377 myScaleX)  (* -0.331 myScaleY))
)
```

- When doing arithmetics for coordinates, where several terms are involved, then use only addition between the terms, not subtraction. The minus cution in LISP “(- ...)” shall be used to flip the sign of terms as needed before adding them. Addition is easier to grasp for the human brain, the math operation IS addition.
- If several signed combinations of terms are needed, then use a redundant “(+ ...)” for positive terms, to increase readability,. This way, only the + or – signs will differ from row to row:

```
(+ (+ thisTerm) (- thatTerm) (+ otherTerm) (- yetAnotherTerm))
(+ (- thisTerm) (- thatTerm) (- otherTerm) (- yetAnotherTerm))
```

- A sketch with **dimensions** shall be included in the leaf routine where the graphic is actually drawn. Routines calling the leaf routine may have sketches without dimensions.
- Numbered coordinate points used for drawing graphics shall be named as p1, p2, etc. They shall be accompanied by a sketch indicating by number (without the 'p') where each point is.
- Coordinates for text positions used in graphics shall be named as t1, t2, etc. They must be accompanied by a sketch indicating by number, with the 't' where each text is, as 't1', 't2, etc.. Exception: If there is already a 'p' point, e.g. p3, at the same coordinate as a text item, e.g. t5, then write 't5=3' and let the '3' occupy the position it represents in the sketch, or make a comment such as “Text 'M' at t1 = p1”
- Comments to the sketch should be place to the right of the sketched, preceded by a semicolon ';' (to make it clear that the comment is not part of the result).
- '.' (period) in a sketch indicates the position of the origin (0,0), as it will be after the whole drawing routines has terminated.
- Sketches shall not include TAB, only SPACE, in addition to printable characters. This improves readbility in editors using TAB setting other than 4.
- **Example:** Sketch at a level above the lowest drawing level – no dimensions included, just showing the expected result. This makes it very easy to browse quickly through a LISP file, looking for your routine without even knowing the routine’s name:

```
;
;
;      ( M )
;      |
;      |
; .
```

- **Example:** Sketch at the lowest drawing level. Dimensions shall be included, showing the detailed expected result. This makes it easy to understand what the subsequent code does, without reading that code in any detail. Note that we had to use '+' at corners to be able to place the '2' centered vertically on its line:

```
;      +---+
;      /   \
;      ( 1 ) ; Text 'M' at t1 = p1
;      \   / ; Radius r
;      +-2-+
;      |
;      3
;      |
;      4
; .
;
; |<---->|      dist
```

- The name of a drawing routine shall be «(defun DrawXxxx (../..)» only whenever it is REALLY generic, same for any administration.
- When a certain drawing routine is specific to one single administration, then its name shall be prefixed by the administration's abbreviation, separated by underscore, as in:

«NOBN_DrawMotor (../..)»

- Always leave three empty rows (just the ENTER) after a «(defun...)» function declaration. This increases readability when browsing through the file.
Exception: The very last function declaration in any LISP file shall be followed by ONE empty row. This makes it possible to move quickly to the bottom when doing copy-paste.
- The 2D symbol's block names may contain non-ASCII characters, but they must be given as AutoCAD UNICODE characters using "(strcat "wordStart" _OSLASH_ "wordEnd")" constructions.
- **Example:** Layer Description for «manøvermaskin» did not display correctly in the AutoCAD Layer Manager after building the 2D library using VLIDE:



- Layers that must be contained in the 2D symbol library – for instance holding sub-parts of a symbol's graphics, so the user may switch it on or off without suppressing the object's layer – must be declared and created from LISP at library generation time.
- **Example:** layDef_HighVoltageSwitchActuator was declared but not created nor used by the LISP code – it is needed to switch on/off the circle-with-an-M in customer drawings where the motor is present in tables but does not show in graphics.
- Whenever one adm needs a new layer or object, please take the hassle of translating to other administrations right away, at least if the object is thought to be in use by everyone. Our professional translators work in ResX, but are not (yet) allowed to work in our .lsp files since the coupling to the XML files is so tight.
- If you create the 2D library from Visual Studio with LISPSYS=1 set before you closed your previous AutoCAD session, then non-ASCII characters created as UTF-8 in Notepad++ are ok.
- If you create the 2D library from VLIDE then LISPSYS=0 must have been set before you closed our previous AutoCAD session. Then non-ASCII must be given as AutoCAD MBCS (multi-byte character sequence) using AutoCAD's UNICODE numbering. There are about 150.000 glyphs present in the UNICODE system. UTF-8 and AutoCAD UNICODE addresses many of them.
- The AutoCAD MBCS set has been declared by us as symbolic constants:
...\\GitHub\\RailCOMPLETE-ALL-2D-LIBRARIES_SRC\\Utilities**CAD system constants.lsp**
- However, we still struggle – I had to give up on this one today and replace the O-slash'es with plain "O" and "o":

```

layDef_HighVoltageSwitchActuator
(list
  (list ; XXGL
    (strcat (ModifyAdmString _OCS_) "$$POWER_SWITCH_ACTUATOR")
    _colorMetaDataTable_
    (strcat _PREFIX_ " " _OCS_NAME_ " - Switch actuator symbol graphics
    _ByBlock_
  )
  (list ; NOBN
    (strcat (ModifyAdmString _OCS_) "$$MAN" _uOSLASH_ "VERMASKIN")
    _colorMetaDataTable_
    (strcat _PREFIX_ " " _OCS_NAME_ " - KL-bryter man" _OSLASH_ "vermas
    _ByBlock_
  )
  (list ; FRSR
    (strcat (ModifyAdmString _OCS_) "$$MOTEUR_INTTERUPTEUR_CATENAIRE")
    _colorMetaDataTable_
    (strcat _PREFIX_ " " _OCS_NAME_ " - moteur interrupteur cat" _EACUT
    _ByBlock_
  )
  (list ; DEDB
    (strcat (ModifyAdmString _OCS_) "$$OBERLEITUNGSTRENNSCHALTER_MASCHI
    _colorMetaDataTable_
    (strcat _PREFIX_ " " _OCS_NAME_ " - Oberleitungstrennschaltermaschi
    _ByBlock_
  )
  (list ; JPTX
    (strcat (ModifyAdmString _OCS_) "$$SWITCH_ACTUATOR")
    _colorMetaDataTable_
    (strcat _PREFIX_ " " _OCS_NAME_ " - Switch actuator symbol graphics
    _ByBlock_
  )
)
)

```

- Example: A typical error message from VLIDE when your code tries to set a layer that has been declared but not yet created (due to an omission).
«(SetLayer layer_LayerNameTHatDoesntExistYet)» fails, our code tries to put the item on a default “catch” layer instead:

```

[6.180] (SETLAYER ((0" 7 "RailCOMPLETE Common - Default layer" " _ByBlock") (0" 7 "RailCOMPL
[7.174] (CREATELAYER ((0" 7 "RailCOMPLETE Common - Default layer" " _ByBlock") (0" 7 "RailCOMPL
[8.168] (SETLAYER ((0" 7 "RailCOMPLETE Common - Default layer" " _ByBlock") (0" 7 "RailCOMPL
[9.162] (DRAWCIRCLEATPOS ((0" 7 "RailCOMPLETE Common - Default layer" " _ByBlock") (0" 7 "RailCOMPL
[10.153] (DRAWMOTOR "U+00C5PEN") LAP+134
[11.147] (NORN-OCS-SWITCH-ACTUATOR) LAP+10

```

Det manglet et kall i 0

- Find AutoCAD's UNICODE MBCS here:

CAD system constants.lsp x NO-BN-OcsMasts.xml x NO-BN-OcsVariousObjects.xml x NO-BN-OcsSwitchesAndTransformers.xml

```

15 (setq
16   ; CAD system selector '_CAD_'.
17   ;=====
18   _ACAD_      "_acad_"    ; Autodesk / AutoCAD and Autolisp compatible products
19   _BCAD_      "_bcad_"    ; Hexagon / Bricsys / BricsCAD. Not implemented as of 2020-04-06
20   ;=====
21   _CAD_       _ACAD_      ; CAD SYSTEM SELECTION HERE! YOU MAY NOW TEST ON '
22   ;=====
23 )
24
25 (defun DefineCadSystemConstants ( / )
26   (cond
27     ((= _CAD_ _ACAD_)
28      ; IMPORTANT NOTES on AutoLISP intricacies:
29      ;
30      ; 1)
31      ; Integer division such as (/ 3 2) returns 1, whereas (/ 3 2.0) and (/ 3.0
32      ;
33      ; 2)
34      ; Note: Unicode hex 'nnnn' is '\U+nnnn' in AutoCAD text - but in later versions
35      ; non-English characters don't work at all. Use globals defined in the global
36      ; (eval (strcat "Skj" _OSLASH_ "testykke")) where you need to use the combination
37      ;
38      ; 3)
39      ; From AutoCAD 2022 and later, the LISPSYS system variable controls the interpretation
40      ; also affects how characters are treated. The LISPSYS variable should be set in the
41      ; MAIN(_) in the 99_Main.lsp file. This should conveniently be set in your
42      ;
43      ; LISPSYS=0
44      ; Visual LISP IDE (VL IDE) is set as the default editor, however AutoLISP files
45      ; AutoLISP source (LSP) files when saved and compiled use the ASCII (MBCS)
46      ; Note: This setting results in the behavior of AutoCAD 2020 and earlier releases
47      ;
48      ; LISPSYS=1
49      ; Visual Studio (VS) Code is set as the default editor and AutoLISP files
50      ; AutoLISP source (LSP) files, when saved, use the encoding set in VS Code,
51      ;
52      ; LISPSYS=2
53      ; Visual Studio (VS) Code is set as the default editor and AutoLISP files
54      ; AutoLISP source (LSP) files, when saved, use the encoding set in VS Code,
55      ;
56      ; We cannot use special national characters in folder names in a 'findfile'
57      ; It also appears that Windows substitutes 'oe' for Scandinavian 'ø' in folder
58      ; avoid 'oe' as well (in folder names). (Discovered 2020-04-06 by CLFEY on
59      ;
60      ; 3)
61      ; AutoLISP ignores overloaded commands when a dot "." precedes the command
62      ;
63      ; 4)
64      ; AutoLISP interprets the command and its parameters according to the US-English
65      ; precedes the command name or attribute name.
66      ;
67      ; 5)
68      ; Subcommand names such as "Justify" in the ATTDEF command will lead to an
69      ; "_J") and you're fine when writing LISP. This is NOT very consequently precise
70      ; instance '(command ". _MOVE" "_AL" "" "0,0" "0,1")' (move everything right)
71      ; abbreviation "AL" ("_AL") and the full text "ALL" (_selectAll_). But (command
72      ; "prompt" "default_text" "Justify" _middleCenter_ "0,0" 3.5 0) (middle center
73      ; letters, 0 deg rotation) does NOT work. The ATTDEF command requires you to
74      ; highlighted when using the command-line version command, i.e. "J" ("_J")
75      ; The same goes for "ARC" "CE"..." which works, but "ARC" "CEnter"..." does

```

Claus Feyling

Daglig leder
CEO

Railcomplete AS

Visit us: Fornebuveien 33, 1366 Lysaker, Norway

Write to us: Brageveien 4a, 0358 OSLO, Norway

claus.feyling@railcomplete.com +47 - 908 24 018

www.railcomplete.com



CONFIDENTIALITY, DISCLAIMER and TRADEMARK NOTICE: This message is for the sole use of the intended recipients and may contain confidential information. If you are not an intended recipient, you are requested to notify the sender by reply e-mail and destroy all copies of the original message. Any unauthorized review, use, disclosure or distribution is prohibited. While the sender has taken reasonable precautions to minimize the risk of viruses, we cannot warrant the absence of, or accept liability for, any such viruses in this message or any attachment. "RailCOMPLETE" and the figurative logo of RailCOMPLETE are registered trademarks of Railcomplete AS, NO916118503, Norway.