

AutoLISP で取得した図形の情報を書き換える

TEXT、CIRCLE、ARC、LINE、LWPOLYLINE、ELLIPSE などの図形が持つ DXF データリストを取得する方法は、既に説明しましたが、今回はこの情報を書き換えることで図形そのものを書き換える方法について説明します。

1. 基本的な手順

図形を書き換える手順は、以下の通りです。

- ① 図形情報（DXF データリスト）を取得する。
- ② 取得した情報（DXF データリスト）の一部（書き換えたい部分のグループコードの値）を希望する内容に書き換える。
または、省略されているグループコードのドットペアを追加する。
- ③ entmod 関数を用いて、AutoCAD 内の図形情報（DXF データリスト）を書き換えることで図形が書き換えられる。

2. 例1 LINEの太さを変更する

① entsel にてLINEを選択、その 図形名を取得する

```
コマンド: (setq linename (car (entsel)))  
オブジェクトを選択: <図形名: 22173a12520>
```

② entget にて 図形名から LINEのDXFデータリスト(エンティティリスト)を取得する

```
コマンド: (setq props (entget linename))  
((-1 . <図形名: 22173a12520>) (0 . "LINE") (330 . <図形名: 22173a101f0>) (5 . "2CA") (100 . "AcDbEntity") (67 . 0) (410 . "Model") (8 . "0") (100 .  
"AcDbLine") (10 2013.92 1411.08 0.0) (11 2441.48 1624.72 0.0) (210 0.0 0.0 1.0))
```

③ 線の太さを決定するグループコード 370 を探す

グループコード 370 は、省略されています。（線の太さは、ByLayer の場合省略可能です。）

④ グループコード 370 を追加する

サブクラスマーカー (100 . "AcDbEntity") に続く部分に、(370 . 50) を追加してみましょう。（詳細は、あとで説明します。）

```
コマンド: (split-list-at (cons 100 "AcDbLine") props)  
コマンド: (setq newprops (append before (list (cons 370 50)) after))  
((-1 . <図形名: 22173a12520>) (0 . "LINE") (330 . <図形名: 22173a101f0>) (5 . "2CA") (100 . "AcDbEntity") (67 . 0) (410 . "Model") (8 . "0") (370 . 50)  
(100 . "AcDbLine") (10 2013.92 1411.08 0.0) (11 2441.48 1624.72 0.0) (210 0.0 0.0 1.0))
```

2. 例1 LINEの太さを変更する（つづき）

⑤ entmod関数を用いて、DXFデータリスト(エンティティリスト)を書き換える

```
コマンド: (entmod newprops)
((-1 . <図形名: 22173a12520>) (0 . "LINE") (330 . <図形名: 22173a101f0>) (5 . "2CA") (100 . "AcDbEntity") (67 . 0) (410 . "Model") (8 . "0") (370 . 50)
(100 . "AcDbLine") (10 2013.92 1411.08 0.0) (11 2441.48 1624.72 0.0) (210 0.0 0.0 1.0))
```

これで、選択したLINE オブジェクトの線の太さは、ByLayer から 0.50mm に変更されました。

※ 途中使用した split-list-at 関数

```
コマンド: (split-list-at (cons 100 "AcDbLine") props)
```

この関数は、下記コードからなる自作の関数で あらかじめコマンドラインに流し込むことによって定義しておいたものです。

その機能は、引数 lst (リスト) を 同じく引数 item (要素) の前と後の2つのリスト before と after に分割するもので、具体的には、LINEオブジェクトから取得した DXFデータリスト (エンティティリスト) を、サブクラスマーカー (100 . “AcDbLine”) より前の部分 before(リスト) と、あとの部分 after(リスト)に分割しています。

そして、entmod に引数として渡したリストは、before + (370 . 50) + after となっているのです。

```
(defun split-list-at (item lst / found)
  (setq before '() )
  (setq after lst)
  (setq found nil)
  (while
    (and
      after
      (not found))
    )
    (if
      (equal (car after) item)
      (setq found T)
      (progn
        (setq before (append before (list (car after))))
        (setq after (cdr after))
      )
    )
  )
  (princ)
)
```

【補足1】 省略される可能性のある サブクラス：AcDbEntity（エンティティ共通部分のDXFデータ）のグループコード

グループコード	内容	省略可能な場合	例	値	
62	色番号	BYLAYER の時	(62 . 1)	色番号 = 1(red)	注1)
420	24 ビット色値	色が既定の色番号で表現可能な場合	(420 . 16564615)	R = 252、G = 193、B = 135	注2)
430	色名	カラーブックを用いていない場合	(430 . "DIC COLOR GUIDE(R)\$DIC 7")	カラーブック「DIC COLOR GUIDE(R)」の カラー「DIC 7」	
6	線種名	BYLAYER の時	(6 . "Continuous")	線種名 = "Continuous"	
440	透明度	0 の時	(440 . 33554559)	透明度 = 50%	注3)
48	線種尺度	1 の時	(48 . 2.0)	線種尺度 = 2	
60	オブジェクトの可視性	0 の時	(60 . 1)	可視性 = 1(不可視)	注4)
370	線の太さ	BYLAYER の時	(370 . 20)	線の太さ = 0.20mm	注5)
390	印刷スタイルへのハードポインタ				注6)

注1) BYLAYER = 256、BYBLOCK = 0、負の数は非表示画層を示す。

注2) 16564615 = FCC187 → R = FC = 252、G = C1 = 193、B = 87 = 135

注3) 33554559 = 0200 007F → 007F = 127 → $127 \div 255 \approx 50\%$

注4) エンティティ単独では可視性は持ちません。ブロック参照の1つの要素としてのエンティティになった場合に可視性を持ちます。
※ 複数のブロック参照で可視性の状態が異なる設定の場合、不整合が起きそうに思うのですが大丈夫なのだそうです。
理屈については、理解できていません。(申し訳ない。)

注5) 線の太さは常にミリメートルが単位であり、作画尺度とは相関を持ちません。

注6) 印刷スタイルには大きく2つに分けると、CTB (色従属印刷スタイル) と STB (名前の付いた印刷スタイル)とがあります。
CTB：レイヤーごとに色を設定し、その色で印刷プロパティを制御する。
STB：レイヤーごと、またはオブジェクトごとに印刷スタイルを割り当てる。
一般的には、CTBが多く使われていますが、より高度な印刷設定を行う場合にSTBを使うことがあるようです。
グループコード 390 は、この STB を使用して、オブジェクトに印刷スタイルを割り当てた場合に発生します。
(経験したことがないので、詳しいことには言及できません。ごめんなさい。)

【補足2】 ActiveX を使って プロパティの値を書き換える

① vlax-dump-object と vla-get-

(vlax-dump-object (vlax-ename->vla-object (car (entsel)))) T) を実行して、LINE を選択してみましょう。

```
コマンド: (vlax-dump-object (vlax-ename->vla-object (car (entsel)))) T
オブジェクトを選択: ; IAcadLine: AutoCAD Line インタフェース
; プロパティの値:
;   Angle (R0) = 3.97279
;   Application (R0) = #<VLA-OBJECT IAcadApplication 00007ff79c0aa058>
;   Delta (R0) = (-54.7212 -59.9777 0.0)
;   Document (R0) = #<VLA-OBJECT IAcadDocument 000001fda042a428>
;   EndPoint = (25564.6 44087.2 0.0)
;   EntityTransparency = "ByLayer"
;   Handle (R0) = "387"
;   HasExtensionDictionary (R0) = 0
;   Hyperlinks (R0) = #<VLA-OBJECT IAcadHyperlinks 000001fdab17ba18>
;   Layer = "画層2"
;   Length (R0) = 81.1895
;   Linetype = "ByLayer"
;   LinetypeScale = 1.0
;   Lineweight = 50
;   Material = "ByLayer"
;   Normal = (0.0 0.0 1.0)
;   ObjectID (R0) = 46
;   ObjectName (R0) = "AcDbLine"
;   OwnerID (R0) = 43
;   PlotStyleName = "ByLayer"
;   StartPoint = (25619.3 44147.2 0.0)
;   Thickness = 50.0
;   TrueColor = #<VLA-OBJECT IAcadAcCmColor 000001fd9316e4c0>
;   Visible = -1
```

```
; サポートされているメソッド:
;   ArrayPolar (3)
;   ArrayRectangular (6)
;   Copy ()
;   Delete ()
;   GetBoundingBox (2)
;   GetExtensionDictionary ()
;   GetXData (3)
;   Highlight (1)
;   IntersectWith (2)
;   Mirror (2)
;   Mirror3D (3)
;   Move (2)
;   Offset (1)
;   Rotate (2)
;   Rotate3D (3)
;   ScaleEntity (2)
;   SetXData (2)
;   TransformBy (1)
;   Update ()
T
```

一気に 選択したLINEオブジェクトのプロパティの一覧を確認することができ、サポートされているメソッドまで出力されます。

しかし、残念なことにこれらを プログラムの引数として使用することはできないので、プログラム内でプロパティの値を使用する場合、各プロパティの名称を用いて 1 つずつ get して変数に入れなければなりません。オブジェクトの線の太さを変数にするときは、以下ようになります。

```
(setq obj (vlax-ename->vla-object (car (entsel))))
(setq lw (vla-get-Lineweight obj))
```

※ ActiveX関数を使用する場合、(vlax-ename->vla-object) にて、DXF図形名は VLAオブジェクトに変換する必要があります。

【補足2】 ActiveX を使うと・・・（つづき）

② vla-put-

（vla-put- 関数を使うと、先ほどの entmod よりも、はるかに簡単に、LINEオブジェクトの 線の太さを変えることができます。
図形名が、ent に代入されているとすると、以下のコードで LINE オブジェクトの線の太さを 0.50mm に変更できます。

```
(setq obj (vlax-ename->vla-object ent))  
(vla-put-Lineweight obj 50)  
(vla-update obj)
```

最終行の（vla-update obj）は、必ずしも必要でないのですが、場合によってはプロパティの値を変更しただけではそれが図に反映されない場合があるように入れてあります。（大量のデータ変更を行う場合などは、最後に1行だけ入れるのもよいと思います。）