

PreCad file format ver 2.3仕様書 release 1

PreCadの特徴

- PreCadは2DCADです。複数のページを作成することができます。
- 各ページには通常のレイヤーの他にシートがあります。シートは縮尺を設定できます。レイヤーもシートも必ず一つあります。
- シートを複数使用することで縮尺の異なる部分図を作成することができます。
- PreCadのファイルは複数のファイルをZIP形式で圧縮し、拡張子をpcadとしたものです。拡張子をzipに変えて展開すると内容を確認することができます。

ファイル構成

- PreCadの図面ファイルはインデックスファイル及び複数の図面ファイルにより構成されています。それらをzipで圧縮後、拡張子をpcadに変更したものです。そのためPreCadのファイルの拡張子をzipに変更し解凍すると構成するファイルを確認することができます。
以下はファイル構成の例です。

```
/
|index
|drawing_1.pcdt
|drawing_2.pcdt
|drawing_3.pcdt
└media/
  | e87099e2-8962-4e33-9de5-87eda5b55270.jpg
  └ 01ed77be-822f-474b-a297-6d4c025ba6be.jpg
```

- インデックスファイル及び図面ファイルはテキストファイルで文字コードはUTF8です。改行コードはLF(0x0A)もしくはCRLF(0x0D, 0x0A)です。
- ファイルフォーマットはタグパラメータ形式を使用しています。タグパラメータ形式はタグの次にパラメータを()で囲み記述します。パラメータは数値、文字列、タグパラメータを複数含めることができます。パラメータの区切りは空白文字です。またパラメータがないタグも存在します。タグの出現順は一部（セクション、ページ、レイヤ、シート、図形の順序など）を除き自由です。

```
tag1(1)           //単独の数値。整数とするか実数とするかはタグの種類による。
tag2("STRING")    //単独の文字列。
tag3(10.0 15.5)   //座標。数値を空白文字で区切る。
tag4(tag1(5.5)tag2("xyz")) //複数のタグパラメータ。
tag5(0xFF000000)  //16進数は整数として解釈する。主に色で使われる。
byLayer()         //パラメータなし。
//次の二つはどちらも同じ（順序は無関係）
tag6(tag1(5.5)tag2(123))
tag6(tag2(123)tag1(5.5))
```

- タグパラメータ形式では数値は整数（Int）もしくは実数（Double）でタグごとに判断します。
- タグ名は大文字小文字を区別します。
- 一部のタグには短縮形があります。この仕様書では短縮形があるものは[]で囲って併記します。
- 改行や空白などは無視されます（ただし、ファイル先頭のfiletypeは除きます）。
- //から行末まではコメントです。
- 文字列はダブルクォーテーションでくくります。文字列中の行末の改行は無視されます。 \ はエスケープ文字です。改行は \n、タブは \t です。 \ の後にその他の文字が来た場合はその文字が使われます（例えばダブルクォーテーションは \" を使います）。
- ファイルの1行目に filetype タグを記入してください。
- たとえば、indexファイルは1行目に filetype("precad_archive") と記入します。これでファイルの種類を判別します。 filetype("precad_archive") はタグパラメータ形式ですが、途中に空白や改行を入れず行の先頭から書いてください（ファイル種別の判別を簡略化するためです）。
- 1行目のfiletypeはアプリケーション側で一行読み込んで文字列として比較しても通常のタグパラメータと同じように解析してもどちらでも問題ありません。
- mediaフォルダは画像などを同梱する時に使います。画像はこのフォルダに保存する以外に各図面ファイル内に文字列として同梱することもできます。

インデクスファイル

インデクスファイルのファイル名は index です。このファイルにページの情報と共通する設定を記入します。

- 1行目に filetype("precad_archive") を空白文字なしで記入してください。
- 2行目以降にfileinfo、contents、settingsのセクションが続きます。この出現順は守ってください。バージョンアップ等で新たにセクションが加わるかもしれませんので未知のセクションは読み飛ばせるようにしてください。

以下はindexファイルの例です。

```
filetype("precad_archive")
fileinfo(
  version(2.0.0)appinfo("precad for windows version 2.0.0.0")
)
contents(
  pages(
    page(
      title("Page1")drawing("drawing_1.pcdt")
    )
    page(
      title("Page2")drawing("drawing_2.pcdt")
    )
  )
)
settings(
  pageIndex(0)
  paper(size(841.0 594.0))
  grid(p0(0.0 0.0)spacing(10.0)div(10)isScaled(0)angle(0.0))
  printInfo(printPaperSize(420.0 297.0)printCenter(0.0 0.0)printScale(2.0))
)
```

fileinfoセクション

ファイルの情報です。

要素	パラメータ	説明
version(version)	version : String	フォーマットのバージョン
appinfo(name)	name : String	作成したアプリの情報

- バージョンは2.1.0のようにメジャーバージョン、マイナーバージョン、リビジョンを小数点で分割します。
- メジャーバージョンは大きなファイル構造の変更時に変更します。マルチページ対応時に1から2に上がりました。メジャーバージョンが上がり旧アプリで読めない場合はエラーを表示します（メジャーバージョンが上がった場合、おそらくそうなるでしょう）。
- マイナーバージョンは要素追加、タグ名の変更・削除などで上がります。この変更で新バージョンのファイルを旧バージョンで読むと情報の欠落などが起こるかもしれません。必要であれば警告を表示します。
- 読み込み時、マイナーバージョンにより処理を変えることを推奨しません。マイナーバージョンはあくまでも参考とし、追加されたタグ（もしくは廃止されたタグ）をもとに読み込みを行うべきです。
- リビジョンはその他軽微な変更で上がります（タグ名のtypo修正など）。

contentsセクション

pages

ページのリストです。1ページから順に並びます。

page

要素	パラメータ	説明
title(title)	title : String	図面のタイトル
drawing(file)	file : String	図面のファイル名

- 図面ファイルの拡張子は pcdt (PreCad Drawing Text)です。

settingsセクション

設定を記述します。

```
settings(  
  pageIndex(0)  
  paper(size(420 297))  
  grid(p0(0 0)spacing(10)div(5)isScaled(0)angle(0.0))  
  printInfo(printPaperSize(420.0 297.0)printCenter(0.0 0.0)printScale(2.0))  
)
```

pageIndex : Int

選択されていたページのインデクス（0が1ページ目）。

paper

用紙設定。画面上で表示される用紙のサイズで印刷用紙で無いことに注意。

要素	パラメータ	説明
size(width height)	width : Double, height : Double	用紙サイズ

grid

グリッドの設定

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	グリッド原点
spacing(s)	s : Double	グリッド間隔
div(n)	n : Int	グリッド間隔をn分割
isScaled(f)	f : Int	0:グリッド間隔は用紙寸 1:グリッド間隔は実寸
angle(s)	s : Double	グリッド角度

printInfo

要素	パラメータ	説明
printPaperSize(width height)	width : Double, height : Double	印刷する用紙のサイズ
printCenter(x y)	x : Double, y : Double	印刷領域の中心座標
printScale(scale)	scale : Double	印刷倍率

図面ファイル

図面ファイルは各ページの図面本体です。

1 行目に filetype("precad_document") を記入してください。これでファイルの種類を判別します。

2 行目以降にfileinfo、contents、settingsのセクションが続きます。この出現順は守ってください。バージョンアップ等で新たにセクションが加わるかもしれませんので未知のセクションは読み飛ばせるようにしてください。

fileinfoセクション

ファイルの情報です。インデックスファイルに準拠します。

要素	パラメータ	説明
version(version)	version : Stgring	フォーマットのバージョン
appinfo(name)	name : String	作成したアプリの情報

contentsセクション

レイヤ、シート、図形を記述します。

drawingファイルでは基本的に使用する要素は事前に定義されていなければいけません。したがって shapesはsheet及びlayerが必要になるのでshapesの前にlayers及びsheetsが必要になります（layersとsheetsの出現順には制約がありません）。

```
contents(  
  layers(  
    layer(name("テスト1")color(0xFF000000)lineWidth(0))  
    layer(name("テスト2")color(0xFF000000)lineWidth(0))  
  )  
  sheets(  
    sheet(name("シート1")scale(0.3))  
    sheet(name("シート2")scale(0.3))  
  )  
  shapes(  
    sheet("シート2")  
    layer("テスト1")  
    Line(...)  
    Circle(...)  
    layer("テスト2")  
    Line(...)  
    ...  
  )  
)
```

layers

レイヤのリストです。並びは表示されるときを重ね順で下から上になります。

sheets

シートのリストです。並びは表示されるときを重ね順で下から上になります。

shapes

図形のリストです。図形以外にsheetおよびlayerタグがあらわれます。

- sheetタグで現在の書き込みシートを切り替えます。
- layerタグで現在の書き込みレイヤを切り替えます。

並びは表示されるときを重ね順で下から上になります。

図形共通

- 座標及び大きさ（線幅、半径、文字サイズ等）は用紙寸です。シートの縮尺で割ることで実寸になります。
- 座標原点は用紙の中心でX軸は右が正、Y軸は上が正です。
- 角度は度(degree)です。左回りが正です。
- 図形には線種など共通する属性があります。
- 省略時の値が書かれていないパラメータもパラメータを省略できますが値は未定義です。

図形属性

lineStyle（線スタイル属性）

線のスタイル。省略時は黒、実線、線幅0です。

```
lineStyle(width(w)color(c)lineType(name)flag(f))
[ 1s ]
```

要素	パラメータ	説明
width(w) [w]	w : Double	線幅。省略時0.0
color(c) [c]	c : Int	線色。省略時0xff000000（黒）
lineType(name) [t]	name : String	線種名。省略時実線
flag(f) [f]	f : Int	フラグ。省略時0

- フラグ
1:ColorByLayer, 0x1000:補助線, 0x2000:TypeByLayer, 0x8000:WidthByLayer

fillStyle（塗りスタイル属性）

円や多角形の塗りのスタイルです。solid（単色）とbyLayerがあります。fillStyleのパラメータにはどちらかを指定してください。fillStyleタグが省略された場合は透明色です。

```
fillStyle(solid(0xFF00FF00))
```

要素	パラメータ	説明
solid(c)	c : Int	面色。省略時、透明（アルファ値が0）
byLayer()		レイヤ色。パラメータはありません。

arrow, startArrow, endArrow（矢印、始点矢印、終点矢印スタイル属性）

矢印のスタイルです。省略時は矢印無し。

```
startArrow(size(3)type(1))
```

要素	パラメータ	説明
size(size) [s]	size : Double	矢印の大きさ
type(id) [t]	id : Int	矢印の種類

- 矢印の種類

ID	種類
0	なし
1	開いた矢印
2	三角形
3	塗りつぶした三角形
4	円
5	塗りつぶした円
6	四角形
7	塗りつぶした四角形
8	斜め線
9	S字

textStyle（文字スタイル）

要素	パラメータ	説明
color(c) [c]	c : Int	文字色。省略時黒
textAlign(a) [ta]	a : Int	文字配置 0:左 1:中央 2:右。省略時0
fontName(name) [fn]	name : String	フォント名
fontHeight(h) [fh]	h : Double	フォントサイズ
fontWidthScale(w) [fw]	w : Double	フォント横幅比。省略時1
fontSpace(s) [fs]	s : Double	文字間隔。省略時0
fontSkewAngle(a) [fa]	a : Double	傾斜角。省略時0
flag(f) [f]	f : Int	フラグ。省略時0

- フラグ
1:ColorByLayer, 0x2000:縦書き, 0x4000:イタリック, 0x8000:ボールド, 0x10_0000:下線,
0x20_0000:取り消し線, 0x80_0000:（予約：折り返し）

markerStyle（点スタイル）

要素	パラメータ	説明
type(name) [t]	name : String	マーカー名
size(s) [s]	s : Double	サイズ

- マーカー名

名称	種類	備考
asterisk	*	
circle	円	
dot	点	sizeを使わない。画面ズームしてもサイズは変わらない。
plus	+	
square	四角	
triangle	三角形	
x	X	

leaderStyle（引出線スタイル）

要素	パラメータ	説明
textOffset(d) [to]	d : Double	文字の下線との離れ。省略時0
textStyle(...) [ts]	文字スタイル参照	文字スタイル
arrowStyle(...) [as]	矢印スタイル参照	矢印スタイル

balloonStyle（バルーンスタイル）

要素	パラメータ	説明
radius(r) [r]	r : Double	半径
textStyle(...) [ts]	文字スタイル参照	文字スタイル
arrowStyle(...) [as]	矢印スタイル参照	矢印スタイル

dimensionStyle（直線寸法スタイル）

要素	パラメータ	説明
extensionLineOffset(d) [eo]	d : Double	引出位置から引出線までの離れ
extensionLineOvershoot(d) [ev]	d : Double	寸法緯からの引出線の延長長さ
dimensionLineExtension(d) [de]	d : Double	寸法緯の延長長さ
textStyle(...) [ts]	文字スタイル参照	文字スタイル
textOffset(d) [to]	d : Double	文字の寸法線との離れ。省略時0
formatStyle(...) [fs]	寸法値フォーマット参照	寸法値フォーマット
tolerance2LinesTextScale(s) [tt]	s:Double	2行の許容差の文字サイズの比率。 省略時0.75
startArrow(...) [sa]	矢印スタイル参照	始点の矢印スタイル。 省略時矢印なし
endArrow(...) [ea]	矢印スタイル参照	終点の矢印スタイル。 省略時矢印なし

radiusStyle（半径寸法スタイル）

要素	パラメータ	説明
dimensionLineExtension(d) [de]	d : Double	寸法緯の延長長さ
textStyle(...) [ts]	文字スタイル参照	文字スタイル
textOffset(d) [to]	d : Double	文字の寸法線との離れ。省略時0

要素	パラメータ	説明
formatStyle(...) [fs]	寸法値フォーマット 参照	寸法値フォーマット
tolerance2LinesTextScale(s) [tt]	s:Double	2行の許容差の文字サイズの比率。 省略時0.75
arrowStyle(...) [as]	矢印スタイル 参照	矢印スタイル

diameterStyle（直径寸法スタイル）

要素	パラメータ	説明
dimensionLineExtension(d) [de(d)]	d : Double	寸法緯の延長長さ
textStyle(...) [ts]	文字スタイル 参照	文字スタイル
textOffset(d) [to]	d : Double	文字の寸法線との離れ。省略時0
formatStyle(...) [fs]	寸法値フォーマット 参照	寸法値フォーマット
tolerance2LinesTextScale(s) [tt]	s:Double	2行の許容差の文字サイズの比率。 省略時0.75
startArrow(...) [sa]	矢印スタイル 参照	始点の矢印スタイル。 省略時矢印なし
endArrow(...) [ea]	矢印スタイル 参照	終点の矢印スタイル。 省略時矢印なし

formatStyle（寸法値フォーマット）

要素	パラメータ	説明
fractionDigits(d) [fd]	d : Int	小数点桁数。省略時1

要素	パラメータ	説明
usesGroupingSeparator(f) [gs]	f : Int	0:区切りなし 1:桁区切り。省略時0
prefix(s) [p]	s : String	前置文字。省略時文字無し
suffix(s) [s]	s : String	後置文字。省略時文字無し

tolerance（許容差）

要素	パラメータ	説明
toleranceType(t) [tt]	t : Int	許容差タイプ 0:無し 1:一段 2:二段。省略時0
value0(v) [v0]	v : Double	許容差上、一段の時使用
value1(v) [v1]	v : Double	許容差下。一段の時は省略
fractionDigits(d) [fd]	d : Int	小数点桁数。省略時1
suffix(s) [s]	s : String	後置文字。省略時文字無し

図形要素

Line（線）

要素	パラメータ	説明
pp(x0 y0 x1 y1)	x0 : Double, y0 : Double, x1 : Double, y1 : Double	始点と終点
p0(x y)	x : Double, y : Double	始点
p1(x y)	x : Double, y : Double	終点
lineStyle(...)	線スタイル 参照	線スタイル。省略時、黒実線、幅0

要素	パラメータ	説明
[ls]		
startArrow(...) [sa]	矢印スタイル 参照	始点の矢印スタイル。省略時矢印なし
endArrow(...) [ea]	矢印スタイル 参照	終点の矢印スタイル。省略時矢印なし

Polyline（連続線）

要素	パラメータ	説明
vertices(x1 y1 x2 y2 x3 y3 ... xn yn) [vs]	x1..xn : Double, y1..yn : Double	頂点
isClosed(flag) [ic]	flag : Int	0:Open 1:Close 省略時0
lineStyle(...) [ls]	線スタイル 参照	線スタイル。省略時、 黒実線、幅0
fillStyle(...) [fs]	塗りスタイル 参照	塗りスタイル。 省略時塗りなし
startArrow(...) [sa]	矢印スタイル 参照	始点矢印。省略時矢印なし
endArrow(...) [ea]	矢印スタイル 参照	終点矢印。省略時矢印なし

- isCloseが1の時は矢印スタイルは使われません。

Spline（スプライン曲線）

要素	パラメータ	説明
vertices(x1 y1 x2 y2 x3 y3 ... xn yn) [vs]	x1..xn : Double, y1..yn : Double	頂点
isClosed(flag)	flag : Int	0:Open 1:Close 省略時0

要素	パラメータ	説明
[ic]		
lineStyle(...) [ls]	線スタイル参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル参照	塗りスタイル。省略時塗りなし
startArrow(...) [sa]	矢印スタイル参照	始点矢印。省略時矢印なし
endArrow(...) [ea]	矢印スタイル参照	終点矢印。省略時矢印なし

- カーディナルスプラインでテンションは0.5です。
- isCloseが1の時は矢印スタイルは使われません。

Bezier（ベジエ曲線）

要素	パラメータ	説明
vertices(x1 y1 x2 y2 x3 y3 ... xn yn) [vs]	x1..xn : Double, y1..yn : Double	頂点と制御点
lineStyle(...) [ls]	線スタイル参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル参照	塗りスタイル。省略時塗りなし
startArrow(...) [sa]	矢印スタイル参照	始点矢印。省略時矢印なし
endArrow(...) [ea]	矢印スタイル参照	終点矢印。省略時矢印なし

- 3次ベジエ曲線です。頂点、制御点 1、制御点2, 頂点の順に並べてあります。頂点数をmとした場合、制御点と頂点の合計nは $n=3*m+1$ になります。
- isCloseが1の時は始点と終点が直線で繋がれ矢印スタイルは使われません。

Circle（円）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	中心点
radius(r) [r]	r : Double	半径
flatness(f) [f]	f : Double	扁平率。省略時1
angle(a) [a]	a : Double	角度（度）。省略時0
lineStyle(...) [ls]	線スタイル参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル参照	塗りスタイル。省略時塗りなし

Arc（円弧）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	中心点
radius(r) [r]	r : Double	半径
flatness(f) [f]	f : Double	扁平率。省略時1
angle(a) [a]	a : Double	角度（度）。省略時0
startAngle(a) [st]	a : Double	開始角（度）。省略時0
sweepAngle(a) [sw]	a : Double	角幅（度）。省略時90
lineStyle(...) [ls]	線スタイル参照	線スタイル。省略時、黒実線、幅0

要素	パラメータ	説明
fillStyle(...) [fs]	塗りスタイル 参照	塗りスタイル。省略時塗りなし
startArrow(...) [sa]	矢印スタイル 参照	始点矢印。省略時矢印なし
endArrow(...) [ea]	矢印スタイル 参照	終点矢印。省略時矢印なし

- 開始角及び角幅は省略可能ですが省略値は暫定のため省略を推奨しません。
- 角幅は左回りが正、右回りは負です。

Marker（点）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	中心点
angle(a) [a]	a : Double	角度（度）。省略時0
markerStyle(...) [ms]	点スタイル 参照	点スタイル
lineStyle(...) [ls]	線スタイル 参照	線スタイル。省略時、黒実線、幅0

- 線スタイルの線種は無効で常に実線になります。
- フラグで線種がレイヤ線種になっていても実線で表示される。
- フラグが仮線の場合は仮線で表示される（印刷不可の仮点となる）。

Text（文字）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	基準点
width(w) [w]	w : Double	幅
height(h) [h]	h : Double	高さ

要素	パラメータ	説明
text(t) [t]	t : String	文字列
angle(a) [a]	a : Double	角度（度）。省略時0
basis(b) [b]	b : Int	配置基準。省略時0（左下）
textStyle(...) [ts]	文字スタイル 参照	文字スタイル。
fillStyle(...) [fs]	塗りスタイル 参照	文字背景塗りスタイル。省略時塗りなし

- 幅と高さは文字列を囲む幅と高さではなく図形選択時に表示される矩形です。折り返しや背景色の範囲に使う予定でしたが実装できていません（折り返しが難しい）。将来的にフラグでそれらの機能を有効にするかもしれません。
- 配置基準 0:左下, 1:中下, 2:右下, 3:左中, 4:中央, 5:右中, 6:左上, 7:中上, 8:右上

Dimension（直線寸法）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	寸法線始点
p1(x y)	x : Double, y : Double	寸法線終点
direction(dx dy) [d]	dx : Double, dy : Double	寸法線から引出位置への方向単位ベクトル
extensionLine0(d) [e0]	d : Double	始点側引き出し線の長さ
extensionLine1(d) [e1]	d : Double	終点側引き出し線の長さ
text(t) [t]	t : String	文字列
tp(d)	d : Double	文字配置位置 0.0が始点、1.0が終点、0.5が寸法線の中央。省略時0.5

要素	パラメータ	説明
enableAutoDimension(f) [ed]	f : Int	1で寸法値自動。省略時1
lineStyle(...) [ls]	線スタイル 参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル 参照	文字背景塗りスタイル。省略時塗りなし
dimensionStyle(...) [ds]	直線寸法スタイル 参照	直線寸法スタイル
tolerance(...) [to]	許容差 参照	許容差。省略時、許容差なし

- 引き出し線の長さがマイナスの場合、directionの逆向きに引き出し線を伸ばす。

Radius（半径寸法）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	寸法線始点
p1(x y)	x : Double, y : Double	寸法線終点
text(t) [t]	t : String	文字列
tp(d)	d : Double	文字配置位置 0.0が中心、1.0が終点、0.5が寸法線の中央。省略時0.5
enableAutoDimension(f) [ed]	f : Int	1で寸法値自動。省略時1
lineStyle(...) [ls]	線スタイル 参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル 参照	文字背景塗りスタイル。省略時塗りなし
dimensionStyle(...) [ds]	半径寸法スタイル 参照	半径寸法スタイル

要素	パラメータ	説明
tolerance(...) [to]	許容差参照	許容差。省略時、許容差なし

- 始点は円の中心。

Diameter（直径寸法）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	寸法線始点
p1(x y)	x : Double, y : Double	寸法線終点
text(t) [t]	t : String	文字列
tp(d)	d : Double	文字配置位置 0.0が始点、1.0が終点、0.5が寸法線の中央。省略時0.5
enableAutoDimension(f) [ed]	f : Int	1で寸法値自動。省略時1
lineStyle(...) [ls]	線スタイル参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル参照	文字背景塗りスタイル。省略時塗りなし
dimensionStyle(...) [ds]	直径寸法スタイル参照	直径寸法スタイル
tolerance(...) [to]	許容差参照	許容差。省略時、許容差なし

Angle（角度寸法）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	中心
radius(r) [r]	r : Double	半径

要素	パラメータ	説明
startAngle(a) [st]	a : Double	開始角（度）。省略時0
sweepAngle(a) [sw]	a : Double	角幅（度）。省略時90
extensionLine0(d) [e0]	d : Double	始点側引き出し線の長さ
extensionLine1(d) [e1]	d : Double	終点側引き出し線の長さ
text(t) [t]	t : String	文字列
tp(d)	d : Double	文字配置位置 0.0が始点、1.0が終点、0.5が寸法線の中央。省略時0.5
enableAutoDimension(f) [ed]	f : Int	1で寸法値自動。省略時1
lineStyle(...) [ls]	線スタイル参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル参照	文字背景塗りスタイル。省略時塗りなし
dimensionStyle(...) [ds]	直線寸法スタイル参照	寸法スタイル。直線寸法スタイルを使用
tolerance(...) [to]	許容差参照	許容差。省略時、許容差なし

- 引き出し線の長さがマイナスの場合、中心と反対側に引き出し線を伸ばす。

Leader（引出線）

要素	パラメータ	説明
vertices(x1 y1 x2 y2 x3 y3 ... xn yn) [vs]	x1..xn : Double, y1..yn : Double	頂点

要素	パラメータ	説明
text(t) [t]	t : String	文字列
lineStyle(...) [ls]	線スタイル 参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル 参照	文字背景塗りスタイル。 省略時塗りなし
leaderStyle(...) [lt]	引出線スタイル 参照	引出線スタイル

- 文字の下線は頂点に含まない。

Balloon（バルーン）

要素	パラメータ	説明
vertices(x1 y1 x2 y2 x3 y3 ... xn yn) [vs]	x1..xn : Double, y1..yn : Double	頂点
text(t) [t]	t : String	文字列
lineStyle(...) [ls]	線スタイル 参照	線スタイル。省略時、黒実線、 幅0
fillStyle(...) [fs]	塗りスタイル 参照	文字背景塗りスタイル。 省略時塗りなし
balloonStyle(...) [bs]	バルーンスタイル 参照	バルーンスタイル

Group（グループ図形）

要素	パラメータ	説明
shapes(S1()S2()...Sn()) [ss]	S1..Sn:図形	図形のリスト

Path（パス図形）

要素	パラメータ	説明
p(...)		パスリスト
lineStyle(...) [ls]	線スタイル参照	線スタイル。省略時、黒実線、幅0
fillStyle(...) [fs]	塗りスタイル参照	塗りスタイル。省略時塗りなし

パス要素

要素	パラメータ	説明
s(x,y)	x:Double, y:Double	始点
e(flag)	flag:Int	0：閉じない, 1:閉じる
l(x1 y1 x2 y2 ... xn yn)	x1..xn : Double, y1..yn : Double	直線
b(x1 y1 x2 y2 ... xn yn)	x1..xn : Double, y1..yn : Double	3次ベジエ曲線。 2つの制御点と 1 つの頂点の組

- パスはs()から始まること。e()のフラグが1の時は最後の点からまで始点まで直線を引く。

Image（画像図形）

要素	パラメータ	説明
p0(x y)	x : Double, y : Double	中心
width(w) [w]	w : Double	幅
height(h) [h]	h : Double	高さ
angle(a) [a]	a : Double	角度（度）。省略時0

要素	パラメータ	説明
image(bytes) [im]	bytes : Bytes	画像データ
src(file)	file : String	画像データファイルへのパス（フォルダ名含む）。例： src("media\\e1ed77be-822f-4743-a295-6d4c025ba6b0.png") フォルダの区切りの\"\\\"は文字列では2つ必要なことに注意。 ファイルに拡張子は無くてよいが、 あったほうがデバッグが楽になると思います。
type(name) [t]	name : String	画像の種類（jpg, png, bmp） 参考値。省略可。

- 画像の種類はバイナリから判断できるためtypeタグは無くてもよい。今のところtypeタグはデバッグで使うことを想定する。
- imageタグとsrcタグは片方のみ有効。両方あった場合の動作は未定義。
- imageタグを使うとファイルサイズが大きくなりますが画像を一つのファイルにまとめることができるのでアプリケーションによっては実装が容易になります。
- 既存の図面に別の図面をインポートで追加する可能性を考えてsrcに使うファイル名の衝突を避けるためuuidを使うことを推奨します。

settingsセクション

設定を記述します。

```
settings(  
  currentLayer("Lyer1")  
  currentSheet("Sheet2")  
)
```

currentLayer

選択中のレイヤ名

currentSheet

選択中のシート名

補足

ファイルバージョン

- インデックスファイルと図面ファイルのバージョンはそれぞれ違う値にすることもできますが、インデックスファイルのメジャーバージョン2以降はインデックスファイルと図面ファイルは同じものにしようと思います。
- メジャーバージョンはファイル構造が変わって下位互換性が無くなった場合に上げます（例えば、メジャーバージョン1のみに対応するアプリではメジャーバージョン2は読めない）。ただし、新たなタグ要素の追加等でそのタグを読み飛ばすことができる場合はメジャーバージョンを上げずマイナーバージョンを上げます。

バージョンによる違い

- ファイルバージョンは文字列でversionタグを使いますが、旧バージョンでは `fileversion(version : Int)` を使いメジャーバージョンのみでした。
- メジャーバージョン1はページが1つしかなく、pagesタグがない代わりにdrawingタグがあります。また、settingsタグはありません（設定はdrawingファイルにあります）。
以下はインデックスファイルのメジャーバージョン1の例です。メジャーバージョン1では図面ファイル名は `drawing` でした。

```
filetype("precad_archive")
fileinfo(
  fileversion(1)appinfo("precad for windows version 1.0.0.0")
)
contents(
  drawing("drawing")
)
```

- 線図形（Line）の始点と終点はppタグでまとめて書き出しますが、バージョン2.1.0までは始点と終点を別々に以下の書式で記述していました。バージョン2.2.0以降、分けて記述するのは冗長なのでppの書式に変更しました。互換性のため読み込みのみ対応しています。

```
p0(x0,y0)p1(x1,y1)
```

- 頂点リスト（vertices()）はファイルバージョン2.0.0以下では次の書式でした。互換性のため読み込みのみ対応しています。

```
points(P(x0,y0)P(x1,y1)...) //Pは大文字
```

- ベジエ曲線の頂点リストは以下の旧型式(ファイルバージョン2.0.0以下)があります(非推奨)。各頂点とその制御点を分けて記述します。制御点は各頂点の始点側(s())と終点側(e())の制御点を記述します。開いたベジエ曲線の場合は始点の終点側の制御点、終点の始点側の制御点は無視されます。記述が冗長なのでこの形式は互換性のために読み込みのみ対応しています。

```
points(P(x0 y0)P(x1 y1)...)
controlPoints(cp(s(sx0 sy0)e(ex0 ey0)cp(s(sx1 sy1)e(ex1 ey1)))...)
```

- バージョン2.1.0以前の直径寸法スタイルと直線寸法スタイルの矢印は始点終点同じものでした。そのため矢印スタイルのタグは `arrowStyle` でした。
- バージョン2.3.0からImageのsrcタグにファイル名を追加、圧縮ファイル内のmediaフォルダ内に画像を保存します。それ以前はimageタグでbase64で画像のバイナリを同梱していました。

短縮形について

タグ名は内部で使っている変数名を元に決めました。そのため比較的文字数が多くなってしまいました。

タグ名が多少長くてもCPUは十分速くファイルは圧縮するためにそれほど問題にならないと思っていました。実際、ファイルサイズの増加はそれほどでもなく速度も(普通のファイルなら)問題になりませんでした。

しかし、Androidでサイズの大きなファイルを読み込むと速度は大きく落ちました。Androidのディスクアクセスは遅かったのです。しかもGC(ガーベッジコレクション)が発生して遅さに拍車をかけることがわかりました。

そのためバージョン2.2.0で短縮形を設けました。