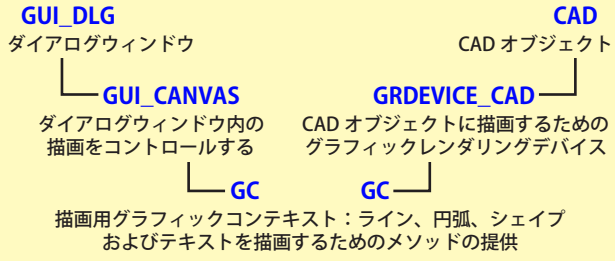
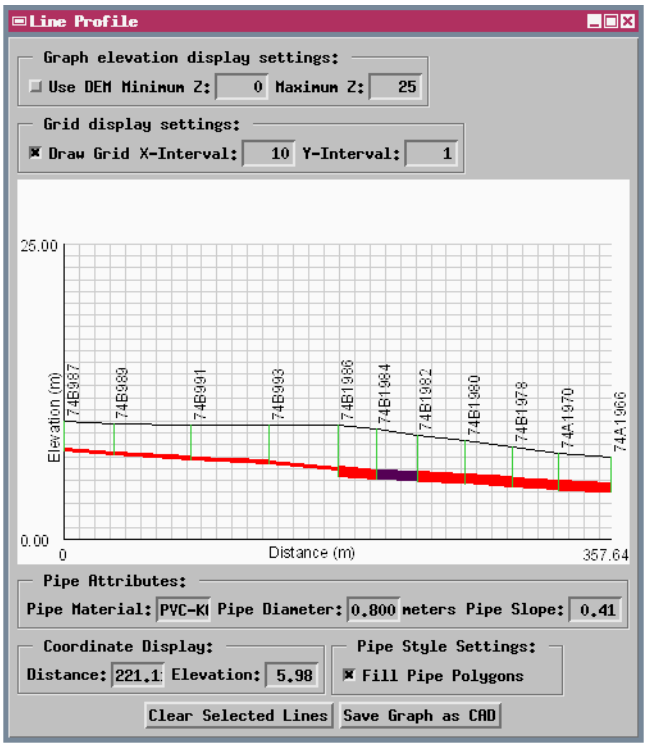
CAD オブジェクトをスクリプトで描く

TNTmips の CAD オブジェクトには地理空間データや地理空間データに関連する座標の無いグラフィック図形、例えば、プロット、断面図、データベーステーブルを図にしたもの、地図の欄外の図などが含まれます。TNT の地理空間スクリプト言語 (SML) を使って図形要素 (ライン、円弧、幾何学図形、およびテキスト) を CAD オブジェクトに書き込むことができます。そうした CAD オブジェクトは地図レイアウト内で使用したり、他のソフトウェアで使うために外部の CAD フォーマットに出力できます。

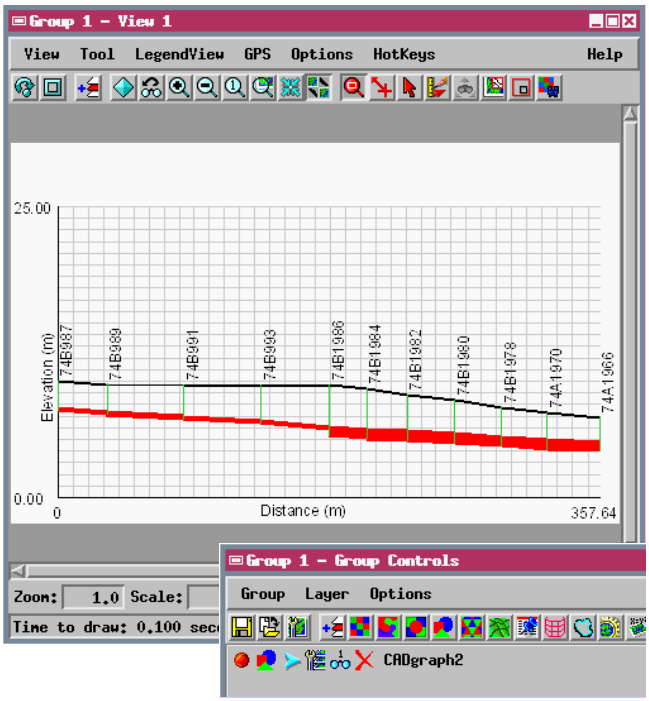
PipeProfileCAD ツールスクリプトでは同じ強力な描画スクリプトを使ってグラフィックスをスクリプトで作成したダイアログウィンドウへ描いたり、複製したり、必要に応じて CAD オブジェクトとして保存することができます。このスクリプトの主な目的は地下の配管網の鉛直データを画面上に表示することです。(テクニカルガイドの「サンプルツールスクリプト：インフラの断面表示 (*Sample Tool Script: Infrastructure Graphical Profile*)」では、このスクリプトを発展させて表示ウィンドウ上にプロットを描いています。) このスクリプトではウィンドウの [グラフを CAD で保存 (Save Graph as CAD)] ボタンを押すと画面表示した断面図を CAD オブジェクトに保存できます。グラフィック要素を描画する一般的方法は GC(グラフィックコンテキスト) クラスによる方法です。それは、ダイアログウィンドウまたは CAD オブジェクトへの出力に関連して使われます。この 2 つの出力先に関するスクリプトのクラスの階層を下図で解説しています。その使用方法が次ページに掲載した描画処理のコードサンプルで比較されています。

ダイアログウィンドウ (左側) と CAD オブジェクト (右側) における描画用地理空間スクリプトのクラス構造
(青字が SML クラス名)





PipeProfileCAD ツールスクリプトで作成した〈ラインプロファイル〉ウィンドウ。地下の配管網の一部の鉛直分布を表示しています。上部の黒線は地表のマンホールを繋いでいます。地下のパイプは赤で描かれています。グラフは表示ウィンドウ内でインタラクティブに選択し連続したライン要素に対するデータベース値を使用して、スクリプトによって描かれています。現在アクティブなライン要素が紫色でハイライトされています。



〈ラインプロファイル〉ウィンドウからパイププロファイルグラフィックを複製した CAD オブジェクトの表示。左図の〈ラインプロファイル〉ウィンドウの [グラフを CAD で保存 (Save Graph As CAD)] ボタンを押すと、選択した新規 CAD オブジェクトにプロファイルが再描画されます。この例では、ツールスクリプトで使用されたのと同じ描画属性、色、フォントを使ってウィンドウグラフィックと CAD オブジェクトの両者を作成しましたが、結果はほとんど同じです。

(翻訳) 株式会社 オープン GIS　東京都墨田区吾妻橋 1-19-14 紀伊国屋ビル 1F
Tel: (03)3623-2851　Fax: (03)3623-3025　E-mail: info@opengis.co.jp

TNT 製品のスクリプト言語の諸機能を解説する多くのサンプルスクリプトが用意されています。これらのスクリプトは www.microimages.com/downloads/tool&macro.htm よりダウンロードできます。

配管断面図 CAD 作成スクリプト (PipeProfileCAD.sml) の抜粋

```
ダイアログウィンドウでグラフ描画に使われる処理：グローバル GC を使用してインタラクティブに要素のハイライトを行う
proc drawGraph() {
   pipeBottomSave = pipeBottom;                          ハイライトしている範囲のパイプを保存する
   pipeTopSave = pipeTop;
   pipeFaceSave = pipeFace;

   createGC();                                           グラフ用のグラフィックコンテキストを作成
   gc.DrawTextSetFont("ARIAL");
   local class COLOR color;

   setTrans(pipeBottom);                                 グラフのアフィン変換を設定する

   local string xlabel = "Distance (m)";                 グラフの軸線をセット
   local string ylabel = "Elevation (m)";
   local numeric drawTwoPointLines = 0;
   local numeric drawStartEndPoints = 1;

   local numeric fontHeight = 12, axisLabelOffset = 3;   グラフのオフセットをセット ( グローバル変数 )
   setGraphOffsets(gc, fontHeight, axisLabelOffset);
                                                         背景に色を塗る
   local class COLOR bgcolor;
   bgcolor.red = 98; bgcolor.green = 98; bgcolor.blue = 98;
   drawBackground(gc, bgcolor);

   color.red = 80; color.green = 80; color.blue = 80;    グリッドを描画
   drawGrid(gc, getGridIntervalX(), getGridIntervalY(), pipeBottom, color);

   color.red = 0; color.green = 0; color.blue = 0;       軸線を描きラベルを付ける
   drawGraphAxes(gc, pipeBottom.GetVertex(pipeBottom.GetNumPoints()-1).x,
      xlabel, ylabel, drawTwoPointLines, color, fontHeight, axisLabelOffset);

   パイプ面を描く
   local class COLOR fill = vectorLayer.SelectedElemColor;
   drawRectangles(gc, pipeFace, color, fill, fillToggle.GetValue());

   class POLYLINE manholeSurfaceLine;                    地表の線を描く (DEM やデータベースから )
   if (doUseDEM()) {
      gc.DrawSetLineStyle("");
      color.red = 0; color.green = 0; color.blue = 0;    DEM の滑らかな地表面ラインを描く
      drawPolyline(gc, smoothedSurface, color);
      manholeSurfaceLine = demSurface;
      }
   else {
      color.red = 0; color.green = 0; color.blue = 0;
      drawPolyline(gc, surface, color);                  地表面ラインを描く
      manholeSurfaceLine = surface;
   }
                        マンホールを描く
   color.red = 20; color.green = 80; color.blue = 20;
   drawManholes(gc, manholeDepth, manholeSurfaceLine, color);

   マンホールのラベルを描く
   color.red = 0; color.green = 0; color.blue = 0;
   drawManholeNames(gc, manholeSurfaceLine, manholeNames, color,
      "ARIAL", 12);

   canvas.Refresh(1);    キャンバスを再描画
}

   CAD オブジェクト内にグラフを描く処理：OnSaveGraph() 処理によって呼ばれた場合、ローカルの GC が渡される
proc drawGraphForCAD(class GC gc) {
   local class COLOR color;
   gc.DrawTextSetFont("ARIAL");

   setTrans(pipeBottom);    グラフのアフィン変換を設定する
```

```
local string xlabel = "Distance (m)";                 グラフの軸線をセット
local string ylabel = "Elevation (m)";
local numeric drawTwoPointLines = 0;
local numeric drawStartEndPoints = 1;

local numeric fontHeight = 12, axisLabelOffset=3;     グラフのオフセットをセット ( グローバル変数 )
setGraphOffsets(gc, fontHeight, axisLabelOffset);
                                                      背景に色を塗る
local class COLOR bgcolor;
bgcolor.red = 98; bgcolor.green = 98; bgcolor.blue = 98;
drawBackground(gc, bgcolor);

color.red = 80; color.green = 80; color.blue = 80;    グリッドを描画
drawGrid(gc, getGridIntervalX(), getGridIntervalY(), pipeBottom, color);

color.red = 0; color.green = 0; color.blue = 0;       軸線を描きラベルを付ける
drawGraphAxes(gc, pipeBottom.GetVertex(pipeBottom.GetNumPoints()-1).x,
   xlabel, ylabel, drawTwoPointLines, color, fontHeight, axisLabelOffset);

パイプ面を描く
local class COLOR fill = vectorLayer.SelectedElemColor;
drawRectangles(gc, pipeFace, color, fill, fillToggle.GetValue());

class POLYLINE manholeSurfaceLine;                    地表の線を描く (DEM やデータベースから )
if (doUseDEM()) {
   gc.DrawSetLineStyle("");
   color.red = 0; color.green = 0; color.blue = 0;    DEM の滑らかな地表面ラインを描く
   drawPolyline(gc, smoothedSurface, color);
   manholeSurfaceLine = demSurface;
   }
else {
   color.red = 0; color.green = 0; color.blue = 0;    地表面ラインを描く
   drawPolyline(gc, surface, color);
   manholeSurfaceLine = surface;
   }

color.red = 20; color.green = 80; color.blue = 20;    マンホールを描く
drawManholes(gc, manholeDepth, manholeSurfaceLine, color);

color.red = 0; color.green = 0; color.blue = 0;       マンホールのラベルを描く
drawManholeNames(gc, manholeSurfaceLine, manholeNames, color,
   "ARIAL", 12);
}


   ダイアログの [ グラフを CAD で保存 ] ボタンが押された時に呼ばれる処理
proc OnSaveGraph() {
                                                      出力する CAD オブジェクトを指定するようユーザにプロンプトを表示
   GetOutputCAD(CADgraph);
         CAD オブジェクトへの描画用 GREDEVICE を作成
   deviceCAD.Create(CADgraph , getHeight(), getWidth() );

   local class GC gcCAD;                              デバイス用のローカルグラフィックコンテキストを作成
   gcCAD = deviceCAD.CreateGC();

   drawGraphForCAD(gcCAD);                            CAD オブジェクトにグラフを描画する処理を呼ぶ

   deviceCAD.Close();                                 CAD グラフィックデバイスを閉じる

   CloseCAD(CADgraph);                                CAD オブジェクトを閉じる
   }
```

(翻訳) 株式会社 オープン GIS　東京都墨田区吾妻橋 1-19-14 紀伊国屋ビル 1F
Tel: (03)3623-2851　Fax: (03)3623-3025　E-mail: info@opengis.co.jp