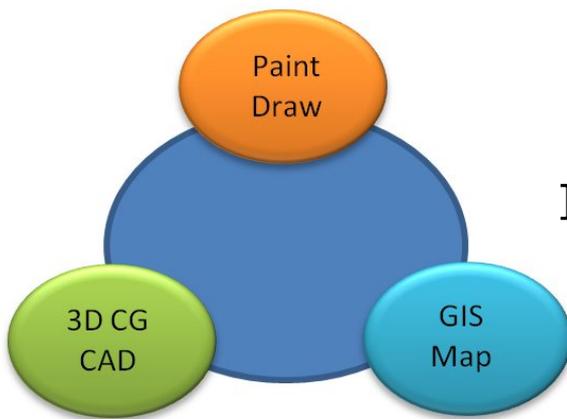


# Delphi Plotter

Version 1.9.0

DCOM (out-of-process server)

ガイド



## 目次

イントロダクション.....	2
DelPlot DCOM オブジェクトツリー 一覧.....	5
コマンドレファレンス .....	6
DelPlotDCOM .....	6
Canvas.....	8
Paper .....	39
Pen.....	41
Font .....	44
Paint .....	46
Shade.....	52
Image.....	55
Trans.....	57
Layers.....	60
Items.....	63
Plugins.....	68
DblPath .....	69
DblMesh .....	71
MeshPlane.....	73
Execute.....	75
Menu.....	93

## イントロダクション

### DelPlot DCOM について

本書は DelPlot DCOM (Automation Server もしくは out-of-process server)機能の利用方法を解説します。この機能により DelPlot をプログラム上から呼び出し、グラフィック ライブラリやフォーマット・コンバーター、描画結果のビューアなど DelPlot の機能を オリジナルアプリケーションの一部として利用を可能にします。

### 本書の構成

本書ではある程度 DelPlot の知識がある方を前提に DelPlot DCOM の各機能の利用方法を記載しています。このため、DelPlot の各画面部品名、描画命令の基本規約、専門用語などについては各説明文に「詳細は描画命令ガイドの方を参照してください」の記載のみが行われています。

### Automaton Server の登録・解除方法

DelPlot DCOM 機能は利用前に Automation Server としてお使いの PC 環境に登録する必要があります。

登録にはコマンドライン (Windows XP では画面上、**スタートボタン**から**ファイル名を指定して実行(R)**)を起動。インストールされた DelPlot を指定して下記を実行します。

```
DelPlot /regserver ↵
```

登録解除には同様にコマンドラインを起動。DelPlot を指定して下記を実行します。

```
DelPlot /unregserver ↵
```

またプログラム環境から DelPlot DCOM の利用にあたって次の設定を行う必要があります。

### Visual Basic

メニューバーより

プロジェクト → 参照の追加 → DelPlot ライブラリ(Ver 2.0)

## Delphi

メニューバーより

**表示 → 登録済みのタイプライブラリ → 登録 → DelPlot\_TLB.pas**

もしくは、下記の添付ファイルをプロジェクトに追加

**DelPlot\_TLB.pas**

概要補足 (DelPlot 規約の一部抜粋)

DelPlot を初めて利用される方がこの DelPlot DCOM を活用されるにあたって、説明文をより円滑に理解する手助けとして DelPlot の特色をマニュアルより抜粋し、列挙します。

### 描画座標のスケール指定

DelPlot に与えるデータは変更可能な描画スケールの単位でその座標系を表現します。描画スケールの単位変更には **Canvas** インターフェースの **ZoomType** プロパティ、画面表示や画像ファイルなどに出力する際の拡大・縮小スケールの変更には **ZoomScale** プロパティを使用します。画面を表示しながら動作を遠隔操作する時は **Execute** インターフェースの **ScaleChange** メソッドを利用できます。

### ペン No、ブラシ No

DelPlot はカルコンプ形式の描画命令方式を継承しているため、線分の描画にはペンという単位でその太さ、線種、色 が多数保管されています。図形の描画にあたってこの内の任意ペンを番号指定してその情報を図形描画に利用します。また、図形の中埋めに関してはブラシという単位でその情報が保管されています。

### 中埋め・影効果について

DelPlot の図形装飾機能には図形へのハッチングの他にグラデーション、影効果を行うことができます。このうちハッチングとグラデーションは共存せず中埋めモードの切り替えで実装されています。中埋めモードの切り替えには **Canvas** インターフェースの **Fill** プロパティから、影効果は同じく **Canvas** インターフェースの **Shade** プロパティから設定を行います。

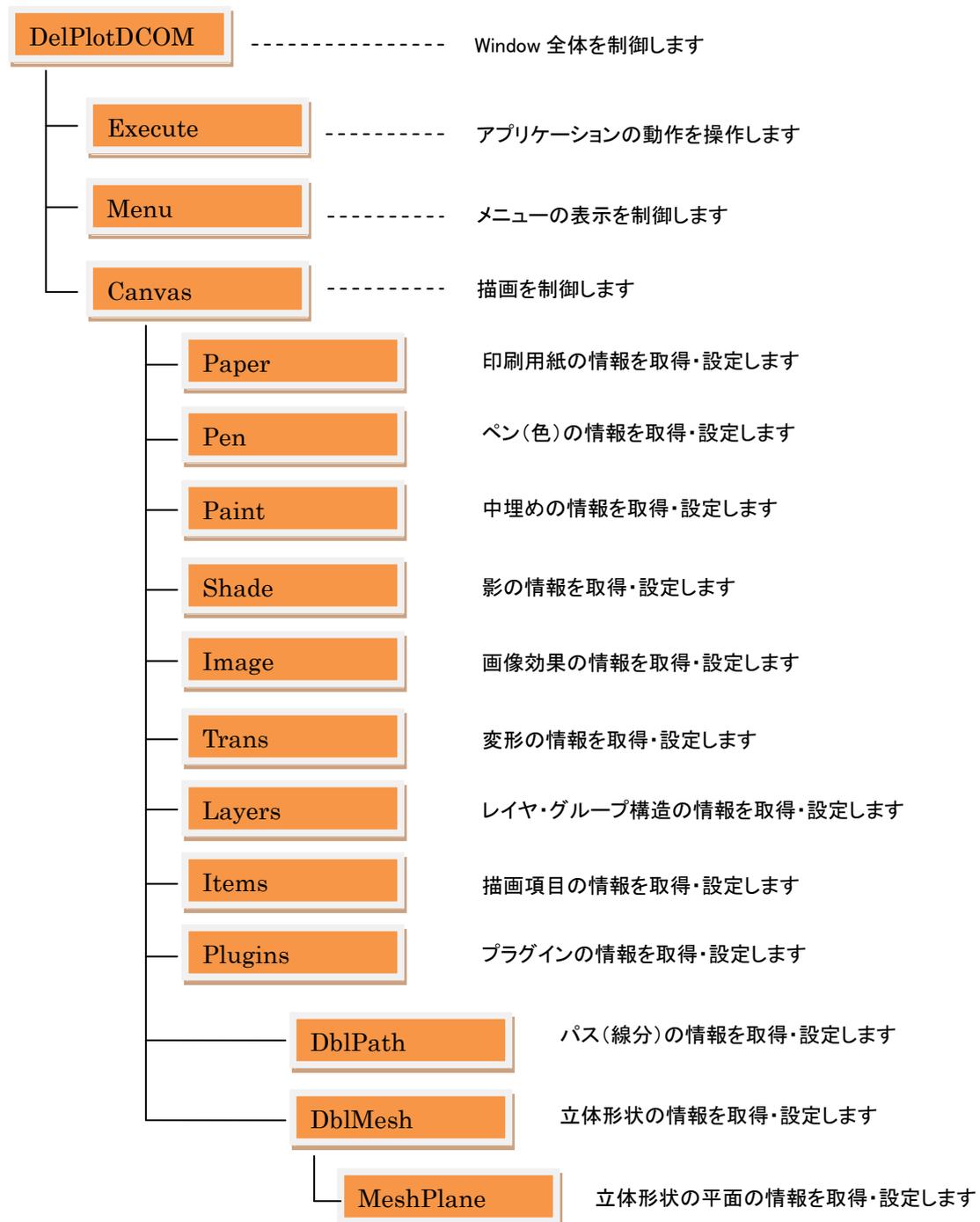
### 拡張文字について

DelPlot の文字描画機能の 1 つに出力対象の文字中に制御文字を埋め込むことで書式を変更したり、特殊なフォントを利用しないと描画できない文字や、テキストで表現が難しい数式（分数やルートなど）を出力する機能があります。

一般的なライブラリにはフォントの太字・斜体への変更がフォント制御機能の中におさまっていますが、DelPlot ではこの拡張文字の中で制御します。

## DelPlot DCOM オブジェクトツリー 一覧

### DelPlot DCOM (Out-of-process Server) Object Map



## コマンドレファレンス

### DelPlotDCOM インターフェース

DelPlot Automation Server オブジェクトツリーのトップに位置するインターフェースです。

#### 使い方

次の使用例は、それぞれの言語での DelPlotDCOM オブジェクトの取得方法です。

#### VisualBasic 6

```
Dim Obj As Object
...
Set Obj = CreateObject("DelPlot.DelPlotDCOM")
```

#### Delphi

```
var
  AutoTestObj : IDelPlotDCOM;
...
AutoTestObj := CoDelPlotDCOM.Create;
```

#### 解説

おもに DelPlot のアプリケーションウィンドウの表示に関する制御を行います。

#### Width プロパティ

表示されるウィンドウの幅サイズを返します。値の設定によってウィンドウサイズの変更も可能です。長整数型 (Long) の値を使用します。

#### Height プロパティ

表示されるウィンドウの高さサイズを返します。値の設定によってウィンドウサイズの変更も可能です。長整数型 (Long) の値を使用します。

---

### Top プロパティ

表示されるウィンドウの位置(Y方向)を返します。値の設定によってウィンドウの配置変更も可能です。長整数型 (Long) の値を使用します。

---

### Left プロパティ

表示されるウィンドウの位置(X方向)を返します。値の設定によってウィンドウの配置変更も可能です。長整数型 (Long) の値を使用します。

---

### Visible プロパティ

ウィンドウの表示状態を返します。値の設定も可能です。論理型 (Boolean) の値を使用します。

---

### Quit メソッド

オートメーション・サーバーを終了します。

### 構文

*expression* .Quit

*expression* 必ず指定します。DelPlotComExport インターフェースを示すオブジェクト式を指定します。

---

### Canvas プロパティ

**Canvas** インターフェースへの参照を返し、DelPlot の描画機能を提供します。値の取得のみ可能です。

### 使用例)

VisualBasic 6

```
Set Obj = CreateObject("DelPlot.DelPlotDCOM")
Obj.Canvas.BeginDoc "CommandTest.plt"
```

## Delphi

```
AutoTestObj := CoDelPlotDCOM.Create;
AutoTestObj.Canvas.BeginDoc("CommandTest.plt", EmptyParam);
```

### Execute プロパティ

**Execute** インターフェースへの参照を返し、DelPlot のウインドウ・コマンドを制御する機能を提供します。値の取得のみ可能です。

### Menu プロパティ

**Menu** インターフェースへの参照を返し、DelPlot のウインドウにあるメニュー項目の表示制御を提供します。値の取得のみ可能です。

## Canvas インターフェース

DelPlot の描画機能と 1 対 1 で対応するオブジェクトです。グラフィックライブラリのように線分や図形を 1 つ 1 つ描いていく事や、DelPlot が持つパラメータの制御を行う事ができます。

### 使い方

他の **Execute**, **Menu** インターフェースの項目がある程度単体で稼働するのに対し、**Canvas** インターフェースの各項目は制御手順を必要とします。利用にあたっては最初に **BeginDoc** メソッドを実行し、それから各操作を実行。一番最後には **EndDoc** メソッドを実行しなければなりません。

**BeginDoc** メソッドに先行し設定された情報はキャンセルされます。

```
○ AutoTestObj := CoDelPlotDCOM.Create;
  AutoTestObj.Canvas.BeginDoc('CommandTest.plt', EmptyParam);
  AutoTestObj.Canvas.AutoSynchronize = true;
  :
```

```
× AutoTestObj := CoDelPlotDCOM.Create;
  AutoTestObj.Canvas.AutoSynchronize = true; ● キャンセルされます
  AutoTestObj.Canvas.BeginDoc('CommandTest.plt', EmptyParam);
```

:

一番最後には **EndDoc** メソッドを実行しなければ描画は完全には終了しません。

```

AutoTestObj.Canvas.BeginDoc('window', true);
:
○ AutoTestObj.Canvas.LineTo(X, Y);
AutoTestObj.Canvas.EndDoc;
AutoTestObj.Execute.SaveToFile("CommandTest.bmp");

```

```

AutoTestObj.Canvas.BeginDoc('window', true);
:
× AutoTestObj.Canvas.LineTo(X, Y);
AutoTestObj. Execute.SaveToFile("CommandTest.bmp");

```

結果が保証されません

**BeginDoc** メソッド, **EndDoc** メソッドについての詳細はそれぞれの解説を参照してください。

## BeginDoc メソッド

描画機能の初期化を行うメソッドです。Canvas オブジェクト内のメソッド、プロパティの指定を行う時は必ず先行して実行する必要があります。

### 構文

構文 1 : *expression*.BeginDoc (*Template*) . . . ファイル・プリンタ指定時  
 構文 2 : *expression*.BeginDoc (*Template* , *Buffer*) . . . 画面出力時  
 構文 3 : *expression*.BeginDoc (*Template* , *PageMax*) . . . PDF ファイル指定時のみ

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*Template* 文字型 (String) の値を使用します。

- ◆ ファイルへの出力時にはファイル名を指定します。ファイル種別はファイル名に含まれる拡張子を検出し、内部で処理分岐します。
- ◆ プリンタへの出力時には文字列 "printer" を指定します。

- ◆ 画面への出力時には文字列 “window” を指定します。

**Buffer** 省略可能です。論理型 (**Boolean**) の値を使用します。画面出力時に、描画情報を保存し、画面上のコマンド(ファイル保存機能など)で活用させたい場合に **True** を指定します。デフォルトは **False** で指定されたコマンドは全て読み捨てされコマンドに活用できません。

**PageMax** 整数型 (**Integer**) の値を使用します。PDF ファイルに描画を保存する時に最大ページ数を指定します。1ページしか出力しない場合は省略可能です。

## 解説

引数 **Template** にファイル名を指定する場合は、パスも含めることができます。

## EndDoc メソッド

描画終了処理を行うメソッドです。**BeginDoc** メソッドと対となり機能し **Canvas** オブジェクトの操作を完結させます。**BeginDoc** メソッドでファイル出力を指定した時はこのメソッドの実行によりファイル出力します。画面出力する時にも **AutoSynchronize** プロパティが **False** に指定されていると、このメソッドが実行されるまで画面表示しません。

## 構文

*expression*.**EndDoc**

*expression* 必ず指定します。**Canvas** インターフェースを表すオブジェクト式を指定します。

## NewPage メソッド

改ページを行います。

## 構文

*expression*.**NewPage**

*expression* 必ず指定します。**Canvas** インターフェースを表すオブジェクト式を指定します。

## 解説

出力対象に複数ページを含む事を認められているフォーマット形式を選択していた時にはこのメソッドを発行した後の描画メソッドは次のページに描画が行われます。出力対象に複数ページを含められないフォーマット形式を選択した時は何も行われません。

---

## TextOut メソッド

テキスト文字列の描画を行います。

### 構文

*expression*.TextOut ( *x*, *y*, *round*, *strValue* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x*, *y* 実数型 (Double) の値を使用します。描画座標を指定します。

*round* 実数型 (Double) の値を使用します。文字列の回転角を指定します。

*strValue* 文字型 (String) の値を使用します。文字列を指定します。

### 解説

文字サイズ、フォント変更は **Font** プロパティで指定します。文字色は現在のペン色を使用します。

Unicode 文字には現在のところ対応していません。

詳細は描画命令ガイド PLOT\_SYMBOL コマンドを参照してください。

描画する文字には拡張書式を使用できます。拡張書式の詳細は描画命令ガイドの方を参照してください。

---

## GraphTextOut メソッド

伸縮などの効果・ハッチングが可能な装飾文字の描画を行います。

### 構文

*expression*.GraphTextOut ( *x*, *y*, *strValue*, *width*, *height*, *round*, *PenNo*, *HatchNo*, *BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

- 
- x, y* 実数型 (Double) の値を使用します。描画座標を指定します。
- strValue* 文字型 (String) の値を使用します。描画する文字列を指定します。
- width, height* 実数型 (Double) の値を使用します。文字列のサイズ(幅、高さ)を指定します。
- round* 実数型 (Double) の値を使用します。文字列の回転角を指定します。
- PenNo* 整数型 (Integer) の値を使用します。文字列の枠線色をペン No で指定します。
- HatchNo* 整数型 (Integer) の値を使用します。文字列の中埋め/ハッチング色をペン No で指定します。
- BackPenNo* 整数型 (Integer) の値を使用します。文字列の背景色をペン No で指定します。

### 解説

TextOut メソッドと異なり、文字サイズは *width, height* が適用されますが、フォント変更は **Font** プロパティで指定します。文字色は現在のペン色を使用します。Unicode 文字には現在のところ対応していません。

詳細は描画命令ガイド GRAPH\_SYMBOL コマンドを参照してください。

描画する文字には拡張書式を使用できます。拡張書式の詳細は描画命令ガイドの方を参照してください。

---

### Polygon メソッド

与えられた点を結んで一連の線で構成された閉じた図形(ポリゴン)を描画します。

### 構文

*expression*.Polygon ( *x, y, PenNo, HatchNo, BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x, y* 実数型 (Double) の一次元配列を使用します。要素点座標を指定します。

**PenNo** 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

**HatchNo** 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。

**BackPenNo** 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

## 解説

詳細は描画命令ガイド PLOT\_POLYGON コマンドを参照してください。

## Polyline メソッド

与えられた点を結んで一連の線で構成された開いた図形(ポリライン)を描画します。

## 構文

*expression*.Polyline (*x*, *y*, *PenNo*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x*, *y* 実数型 (Double) の一次元配列を使用します。要素点座標を指定します。

**PenNo** 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

## 解説

詳細は描画命令ガイド PLOT\_POLYLINE コマンドを参照してください。

## Bezierline メソッド

ベジェ曲線を描画します。

## 構文

*expression*.Bezierline (*x*, *y*, *PenNo*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

- x, y*** 実数型 (Double) の一次元配列を使用します。要素点座標を指定します。
- PenNo*** 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。
- HatchNo*** 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。
- BackPenNo*** 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

### 解説

*x, y* の配列で示される最初の曲線は、1 番目の点と 4 番目の点を結んで描画され、2 番目と 3 番目の点はコントロールポイントとして使われます。シーケンスで後続する次の曲線は必ずさらに 3 個の点を必要とします。先行する曲線の終点は次の曲線の始点として、次の 2 個の点はコントロールポイントとして、3 番目の点が終点として使われます。

詳細は描画命令ガイド PLOT\_BEZIER コマンドを参照してください。

## FreeSurface メソッド

ベジェ曲線で囲まれた閉じた図形を描画します。

### 構文

*expression*.FreeSurface ( *x, y, PenNo, HatchNo, BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

- x, y*** 実数型 (Double) の一次元配列を使用します。要素点座標を指定します。
- PenNo*** 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。
- HatchNo*** 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。
- BackPenNo*** 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

### 解説

$x, y$  の配列で示される最初の曲線は、1 番目の点と 4 番目の点を結んで描画され、2 番目と 3 番目の点はコントロールポイントとして使われます。シーケンスで後続する次の曲線は必ずさらに 3 個の点を必要とします。先行する曲線の終点は次の曲線の始点として、次の 2 個の点はコントロールポイントとして、3 番目の点が終点として使われます。

詳細は描画命令ガイド PLOT\_FREESURFACE コマンドを参照してください。

---

## Rect メソッド

四角形の描画を行います。

### 構文

*expression*.Rect ( *x, y, width, height, round, PenNo, HatchNo, BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x, y* 実数型 (Double) の値を使用します。描画座標 (左下点) を指定します。

*width, height* 実数型 (Double) の値を使用します。図形の幅、高さを指定します。

*round* 実数型 (Double) の値を使用します。図形の回転角を指定します。

*PenNo* 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

*HatchNo* 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。

*BackPenNo* 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

### 解説

詳細は描画命令ガイド PLOT\_RECTANGLE コマンドを参照してください。

---

## RoundRect メソッド

角の丸い四角形の描画を行います。

## 構文

*expression.RoundRect* ( *x*, *y*, *width*, *height*, *round*, *rx*, *ry*, *PenNo*, *HatchNo*, *BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x*, *y* 実数型 (Double) の値を使用します。描画座標 (左下点) を指定します。

*width*, *height* 実数型 (Double) の値を使用します。図形の幅、高さを指定します。

*round* 実数型 (Double) の値を使用します。図形の回転角を指定します。

*rx*, *ry* 実数型 (Double) の値を使用します。四角形を丸く削る丸め幅、丸め高さを指定します。

*PenNo* 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

*HatchNo* 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。

*BackPenNo* 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

## 解説

丸く削れた角の曲線は、*rx*、*ry* の幅、高さの楕円の曲率と一致します。

詳細は描画命令ガイド PLOT\_FREESURFACE コマンドを参照してください。

---

## Ellipes メソッド

楕円の描画を行います。

## 構文

*expression.Ellipes* ( *x*, *y*, *rx*, *ry*, *round*, *PenNo*, *HatchNo*, *BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x*, *y* 実数型 (Double) の値を使用します。図形の中心座標を指定します。

- rx, ry* 実数型 (Double) の値を使用します。楕円の半径 (X、Y 方向) を指定します。
- round* 実数型 (Double) の値を使用します。図形の回転角を指定します。
- PenNo* 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。
- HatchNo* 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。
- BackPenNo* 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

## 解説

詳細は描画命令ガイド PLOT\_ELLIPES コマンドを参照してください。

## Arc メソッド

円弧の描画を行います。

## 構文

*expression.Arc* ( *x, y, rx, ry, round, sra, era, PenNo* )

*expression* 必ず指定します。Canvas インターフェイスを表すオブジェクト式を指定します。

- x, y* 実数型 (Double) の値を使用します。図形の中心座標を指定します。
- rx, ry* 実数型 (Double) の値を使用します。円弧の半径 (X、Y 方向) を指定します。
- round* 実数型 (Double) の値を使用します。図形の回転角を指定します。
- sra, era* 実数型 (Double) の値を使用します。円弧の描画開始角、終了角を指定します。
- PenNo* 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

## 解説

描画開始角、終了角は反時計回りに 0 ~ 360° で指定します。

詳細は描画命令ガイド PLOT\_ARC コマンドを参照してください。

## Pie メソッド

扇型の描画を行います。

### 構文

*expression*.Pie (*x, y, rx, ry, round, sra, era, PenNo, HatchNo, BackPenNo*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x, y* 実数型 (Double) の値を使用します。図形の中心座標を指定します。

*rx, ry* 実数型 (Double) の値を使用します。扇型の半径 (X、Y 方向) を指定します。

*round* 実数型 (Double) の値を使用します。図形の回転角を指定します。

*sra, era* 実数型 (Double) の値を使用します。円弧の描画開始角、終了角を指定します。

*PenNo* 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

*HatchNo* 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。

*BackPenNo* 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

### 解説

描画開始角、終了角は反時計回りに 0 ~ 360° で指定します。

詳細は描画命令ガイド PLOT\_PIE コマンドを参照してください。

## Chord メソッド

弓型の描画を行います。

### 構文

---

*expression.Chord* ( *x, y, rx, ry, round, sra, era, PenNo, HatchNo, BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x, y* 実数型 (Double) の値を使用します。図形の中心座標を指定します。

*rx, ry* 実数型 (Double) の値を使用します。楕円の半径 (X、Y 方向) を指定します。

*round* 実数型 (Double) の値を使用します。図形の回転角を指定します。

*sra, era* 実数型 (Double) の値を使用します。円弧の描画開始角、終了角を指定します。

*PenNo* 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

*HatchNo* 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。

*BackPenNo* 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

## 解説

描画開始角、終了角は反時計回りに 0 ~ 360° で指定します。

詳細は描画命令ガイド PLOT\_CHORD コマンドを参照してください。

---

## Draw メソッド

画像の描画を行います。

## 構文

*expression.Draw* ( *x, y, FileName, width, height, round, rx, ry, style, TransparentPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x, y* 実数型 (Double) の値を使用します。描画座標を指定します。

*FileName* 文字型 (String) の値を使用します。画像ファイル名を指定します。

*width, height* 実数型 (Double) の値を使用します。画像の幅、高さを指定します。

*round* 実数型 (Double) の値を使用します。画像の回転角を指定します。

*rx, ry* 実数型 (Double) の値を使用します。画像の回転中心座標を指定します。

*style* 整数型 (Integer) の値を使用します。画像の表示方法を指定します。

塗りつぶし	0
透過処理	1
シルエット	2

*TransparentPenNo* 整数型 (Integer) の値を使用します。画像の透過色に指定する色  
ペン No で指定します。

### 解説

*style* に 0 が指定されている時は、*TransparentPenNo* は意味を持ちません。

詳細は描画命令ガイド PLOT\_IMAGE コマンドを参照してください。

## LineTo メソッド

直線の描画を行います。

### 構文

*expression*.LineTo ( *x*, *y* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x*, *y* 実数型 (Double) の値を使用します。描画座標を指定します。

### 解説

直線は現在のペン座標から指定された座標までで指定されます。また、描画後にペン座標を指定座標に変更します。

詳細は描画命令ガイド PLOT\_LINE コマンドを参照してください。

## MoveTo メソッド

現在のペン座標を指定座標に変更します。

### 構文

*expression*.MoveTo ( *x*, *y* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x*, *y* 実数型 (Double) の値を使用します。描画座標を指定します。

詳細は描画命令ガイド PLOT\_LINE コマンドを参照してください。

## Path メソッド

直線とベジェ曲線で構成されるパス図形を描画します。

### 構文

*expression*.Path ( *x*, *y*, *command*, *PenNo*, *HatchNo*, *BackPenNo* )

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x*, *y* 実数型 (Double) の一次元配列を使用します。要素点座標を指定します。

*command* 文字型 (String) の値を使用します。要素点の種別をキャラクタで表し、*x*, *y* 配列数分の長さの文字列として指定します。

m	移動
M	移動 (パス閉じる)
l	直線
L	直線 (パス閉じる)
c	ベジェ曲線
C	ベジェ曲線 (パス閉じる)

*PenNo* 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

**HatchNo** 整数型 (**Integer**) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。

**BackPenNo** 整数型 (**Integer**) の値を使用します。図形の背景色をペン No で指定します。

## 解説

詳細は描画命令ガイド PLOT\_PATH コマンドを参照してください。

## Shape メソッド

図形を描画します。種別を指定する事により(矩形、角丸矩形、楕円、円弧、扇形、弓形)を描画します。

## 構文

*expression*.Shape(*x, y, width, Height, Rot, rx, ry, Style, PenNo, HatchNo, BackPenNo*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x, y* 実数型 (**Double**) の値を使用します。図形の中心座標を指定します。

**Width, Height** 実数型 (**Double**) の値を使用します。Width: X 軸方向の半径。Height: Y 軸方向の半径を指定します。

**Rot** 実数型 (**Double**) の値を使用します。図形の回転角を指定します。

*rx, ry* 矩形、楕円: 設定なし。  
角丸矩形: (丸め幅、高さ) 描画する図形の角の丸め幅を指定する。  
それ以外: 描画する図形の描画開始、終了角を 0 度から反時計回りの角度で指定する。

**Style** 整数型 (**Integer**) の値を使用します。描画する図形の種別を指定します。

四角形	0
正方形	1
角丸矩形	2
楕円	3

円	4
円弧	5
扇形	6
弓形	7

**PenNo** 整数型 (Integer) の値を使用します。図形の枠線色をペン No で指定します。

**HatchNo** 整数型 (Integer) の値を使用します。図形の中埋め/ハッチング色をペン No で指定します。

**BackPenNo** 整数型 (Integer) の値を使用します。図形の背景色をペン No で指定します。

### 解説

詳細は描画命令ガイド PLOT\_SHADE コマンドを参照してください。

## Mesh メソッド

立体形状を描画します。

### 構文

*expression*.Mesh(*x, y, z PenNo, HatchNo, BackPenNo*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*x, y z* 実数型 (Double) の一次元配列を使用します。要素点座標を指定します。

### 解説

立体形状の点座標を登録します。点座標を繋いだ平面の登録は登録後、内容の編集する事で行います。その詳細は [DbIMesh インターフェース](#) を参照してください。

その他の詳細は描画命令ガイド PLOT\_MESH コマンドを参照してください。

## Camera メソッド

立体形状 (PLOT\_MESH) をページ上に投影する視点を操作します。

## 構文

*expression*.Camera(*zoom*, *px*, *py*, *pz*, *rx*, *ry*, *rz*, *ox*, *oy*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*zoom* 実数型 (Double) の値を使用します。要素点座標を指定します。

*px*, *py*, *pz* 実数型 (Double) の値を使用します。3D 座標系内での注視点座標を指定します。

*rx*, *ry*, *rz* 実数型 (Double) の値を使用します。3D 座標系内での視点方向を指定します。

*ox*, *oy* 実数型 (Double) の値を使用します。3D 座標系を 2D 面へ投影する座標を指定します。

## 解説

詳細は描画命令ガイド DEFINE\_CAMERA コマンドを参照してください。

## Trans メソッド

描画項目に座標変換を行い変形します。

## 構文

*expression*.Trans(*n*, *m* *Sx*, *Sy*, *Tx*, *Ty*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*n*, *m* 整数型 (Integer) の値を使用します。変形格子の格子数を指定します。

*Sx*, *Sy* 実数型 (Double) の一次元配列を使用します。変形前の図形に設定する格子座標を指定します。

*Tx*, *Ty* 実数型 (Double) の一次元配列を使用します。*Sx*, *Sy* に対応する変形後座標を指定します。

## 解説

詳細は描画命令ガイド DEFINE\_TRANS コマンドを参照してください。

## Include メソッド

外部のデータファイルの内容を「データの追加なく」反映(リンク)します。

### 構文

*expression.Include(Style, strValue)*

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*style* 整数型 (Integer) の値を使用します。データ挿入の種別を指定します。

0	データ(plt)の内容を反映する
1	データ(plt)等の描画結果を画像として取り込む
2	データ入力用プラグインを使用する

*strValue* 文字列 (String) の値を使用します。挿入データを指定します。

### 解説

詳細は描画命令ガイド PLOT\_INCLUDE コマンドを参照してください。

## PlotLabel メソッド

ラベル設定された描画項目、レイヤを描画します。

### 構文

*expression.PlotLabel (LabelNo, x, y, w, h, Rot)*

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*LabelNo* 整数型 (Integer) の値を使用します。描画するラベル番号を指定します。

*x, y* 実数型 (Double) の値を使用します。基準位置からのオフセットを指定します。

*w, h* 実数型 (Double) の値を使用します。元サイズからの拡大率を指定します。

*Rot* 実数型 (Double) の値を使用します。回転角を指定します。

### 解説

詳細は描画命令ガイド PLOT\_LABEL コマンドを参照してください。

## Supply メソッド

DelPlot にプラグインとして認識された利用可能な拡張ツールを使用します。

### 構文

*expression*.Supply (*Style, strValue*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*style* 整数型 (Integer) の値を使用します。0 を指定します。

*strValue* 文字列 (String) の値を使用します。起動するプラグインと利用オプションを指定します。

### 解説

詳細は描画命令ガイド PLOT\_SUPPLY コマンドを参照してください。

## SystemControl メソッド

DelPlot の画面・メニューの表示、システムパラメータを設定します。

### 構文

*expression*.SystemControl (*Style, strValue*)

*expression* 必ず指定します。Canvas インターフェースを表すオブジェクト式を指定します。

*style* 整数型 (Integer) の値を使用します。操作する種別を指定します。

---

**strValue** 文字列 (String) の値を使用します。種別ごとに操作内容を指定します。

### 解説

詳細は描画命令ガイド SYSTEM\_CONTROL コマンドを参照してください。

---

### LayersCount プロパティ

ページ内に登録されているレイヤ・グループ数を返します。値の取得が可能です。長整数型 (Long) の値を使用します。

### 解説

詳細は [Layers インターフェース](#) を参照してください。

---

### ItemsCount プロパティ

ページ内に登録されている描画項目数を返します。値の取得が可能です。長整数型 (Long) の値を使用します。

### 解説

詳細は [Items インターフェース](#) を参照してください。

---

### PluginsCount プロパティ

DelPlot で現在認識されているプラグイン数を返します。長整数型 (Long) の値を使用します。

### 解説

詳細は [Plugins インターフェース](#) を参照してください。

---

### Mask プロパティ

マスク状態を指定します。長整数型 (Long) の値を使用します。

---

**解説**

詳細は描画命令ガイド PLOT\_MASK コマンドを参照してください。

---

**Width, Height プロパティ**

描画範囲の幅、高さを返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

描画範囲の変更は画面・画像ファイル出力時に有効です。プリンタ出力時に指定された場合、設定は無視されます。

DelPlot 描画コマンドに相当する命令を発行するメソッドを指定する前に指定されたものが有効になります。DelPlot 描画コマンドが開始された後で変更されたものは無効です。

詳細は描画命令ガイド DRAW\_SCALE コマンドを参照してください。

---

**DotPitch, DashPitch プロパティ**

ドット、ダッシュの幅長を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

線種(破線、点線、一点鎖線、二点鎖線)を構成するドット、ダッシュの幅長の取得・設定を行います。

詳細は描画命令ガイド LINE\_SCALE コマンドを参照してください。

---

**ZoomType, ZoomGraph, ZoomFont プロパティ**

描画コマンドで指定する数値の画面表示 拡大率、印刷拡大率を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

## 解説

ZoomType プロパティには各スケールの種別 No を指定します。ZoomGraph（図形拡大率）、ZoomFont（文字拡大率）プロパティには自動的に拡大率が計算されセットされます。

単位指定	値
フリースケール	0
ピクセル(Pixel)	1
インチ(Inch)	2
ミリメートル(mm)	3
経緯度（19 座標系： メートル）	4
経緯度（UTM2000）	5

ZoomType プロパティが 0 の時 ZoomGraph、ZoomFont プロパティを設定する事が可能です。0 以外の時に指定された値は無効です。

詳細は描画命令ガイド DRAW\_SCALE コマンドを参照してください。

## OffsetX, OffsetY プロパティ

描画開始位置のオフセットを返します。値の取得・設定が可能です。長整数型（Long）の値を使用します。

## 解説

オフセット値を設定する場合、設定値は現在のオフセット値に**加算**されます。

詳細は描画命令ガイド MOVE\_OFFSET コマンドを参照してください。

## TextAlignX, TextAlignY プロパティ

文字の描画原点位置を返します。値の取得・設定が可能です。長整数型（Long）の値を使用します。

## 解説

TextAlignX プロパティの取得・設定値

左端	0
右端	1
中央	2

**TextAlignY** プロパティの取得・設定値

下端	0
上端	1
中央	2

詳細は描画命令ガイド `TEXT_ALIGNMENT` コマンドを参照してください。

#### AnimationLoop プロパティ

アニメーション表示機能の繰り返し設定を返します。値の取得・設定が可能です。論理型 (`Boolean`) の値を使用します。

#### 解説

全ページをアニメーション表示する機能において最終ページを表示した後、最初のページに戻り、エンドレスに繰り返すかどうかを返します。

詳細は描画命令ガイド オプション画面を参照してください。

#### AnimationTerm プロパティ

アニメーション表示機能の表示間隔を返します。値の取得・設定が可能です。実数型 (`Double`) の値を使用します。

#### 解説

全ページをアニメーション表示する機能においてページの切り替え間隔を秒単位で指定します。

詳細は描画命令ガイド オプション画面を参照してください。

---

**TransparentColor プロパティ**

PNG, GIF ファイルを出力する際の透過色を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

透過色はシステムパレット・カラーの整数値で指定するか、青, 緑, 赤の RGB カラーの輝度で指定します。値 0x00FF0000 は最高の輝度, つまり純粋な青を示し, 0x0000FF00 は純粋な緑を示し, 0x000000FF は純粋な赤を示します。0x00000000 は黒, 0x00FFFFFF は白です。

詳細は描画命令ガイド ファイルオプション画面を参照してください。

---

**Transparent プロパティ**

PNG, GIF ファイルを出力する際に透過処理を行うかどうかを返します。値の取得・設定が可能です。論理型型 (Boolean) の値を使用します。

**解説**

詳細は描画命令ガイド ファイルオプション画面を参照してください。

---

**JpegQuality プロパティ**

JPEG ファイルを出力する際のデータ圧縮率を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

値は 1 ~ 100 の範囲で指定します。

詳細は描画命令ガイド ファイルオプション画面を参照してください。

---

**StrRotDirect プロパティ**

文字の回転方向設定を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

文字の回転方向 取得・設定値

反時計回り	0
時計回り	1

詳細は描画命令ガイド オプション画面を参照してください。

---

**AspectRatioY プロパティ**

描画コマンドで指定される数値を描画する際に乗算される Y 方向のアスペクト値を返します。値の取得・設定が可能です。実数型 (Double) の値を使用します。

---

**PlotPriority プロパティ**

Windows 上において DelPlot 描画処理の優先度を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

優先度の値は 0 ~ 100 の間で設定します。

このプロパティは [BeginDoc](#) メソッドを実行しなくても利用可能です。

詳細は描画命令ガイド オプション画面を参照してください。

---

**AcrobatPattnr プロパティ**

PDF ファイル出力時にハッチングのパターンサイズを印刷用に小さくするかを返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

## 解説

ハッチングのパターンサイズ 取得・設定値

画面用（デフォルト）	0
印刷用スモールサイズ	1

詳細は描画命令ガイド ファイルオプション画面を参照してください。

## CurrentPen プロパティ

現在描画に利用しているペン番号を返します。値の取得・設定が可能です。長整数型（Long）の値を使用します。

## 解説

ペンとは描画に用いるライン情報の事を言います。DelPlot ではこの情報を 255 個持っていて番号（1 ~ 255）で指定する事で利用できます。

詳細は描画命令ガイド コマンド全体規約 CHANGE\_PEN コマンドを参照してください。

## ParameterSave プロパティ

オートメーションサーバーを介して設定した値を DelPlot の次回起動に引き継ぐかどうかを返します。値の取得・設定が可能です。論理型（Boolean）の値を使用します。

## 解説

デフォルトの動作ではオートメーションサーバーとして起動した時に設定したパラメータは、その場限りでアプリケーションとして次回起動した時などには引き継ぎません。設定値を引き継ぐ場合は true にする必要があります。

## PrintAbsZero プロパティ

描画基準点の設定を返します。値の取得・設定が可能です。長整数型（Long）の値を使用します。

## 解説

描画基準点とは座標系の Y 軸を対象に対して左上(↑低、↓高)に取るか、左下(↓低、↑高)を決めるものです。

描画基準点の取得・設定値

左上原点	0
左下原点	1

このプロパティは **BeginDoc** メソッドを実行しなくても利用可能です。

詳細は描画命令ガイド オプション画面を参照してください。

## ClipBordFrom プロパティ

描画結果をクリップボードにコピーするフォーマット形式を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

## 解説

クリップボード出力フォーマット形式の取得・設定値

BMP	0
WMF	1

このプロパティは **BeginDoc** メソッドを実行しなくても利用可能です。

詳細は描画命令ガイド オプション画面を参照してください。

## JustSizeArea プロパティ

描画範囲で生成画像にトリミング(空白削除)を行う設定を返します。値の取得・設定が可能です。論理型 (Boolean) の値を使用します。

## 解説

「データの最大最少を出力サイズにする」での設定を行うものです。画像は描画された範囲のサイズで生成されます。プリンタ出力には無効です。

このプロパティは **BeginDoc** メソッドを実行しなくても利用可能です。

詳細は描画命令ガイド オプション画面を参照してください。

---

#### ZoomScale プロパティ

描画データの拡大率を返します。値の取得・設定が可能です。実数型 (**Double**) の値を使用します。

#### 解説

DelPlot メイン画面で指定する拡大・縮小操作が行われた状態で画像生成を行います。値は**百分率**で指定します。

このプロパティは **BeginDoc** メソッドを実行しなくても利用可能です。

詳細は描画命令ガイドを参照してください。

---

#### AutoSynchronize プロパティ

描画コマンドの実行後、即、画面に描画を反映する設定を返します。値の取得・設定が可能です。論理型 (**Boolean**) の値を使用します。

#### 解説

**BeginDoc** メソッドにおいて画面出力 (Window) を指定した時 有効です。デフォルトの動作では **EndDoc** メソッドを実行して初めて画面に描画結果が反映されます。

しかし、画面上の機能は **EndDoc** メソッドが実行されて描画が完結しない限り利用不可です。

---

**ExtendText プロパティ**

拡張文字の解析・出力機能を開始・停止します。値の取得・設定が可能です。論理型 (**Boolean**) の値を使用します。

**解説**

DelPlot の拡張文字出力機能には若干ながら構文解析に時間が費やされています。DelPlot で高速に大量に文字出力を行いたいけど別に特殊な装飾が必要ない場合に、この解析時間をスキップさせ描画時間の短縮を図ることができます。

---

**InputFileEncode プロパティ**

入力ファイルの文字コードを指定します。長整数型(**Long**) の値を使用します。

**解説**

DelPlot の入力ファイルに使用されている漢字コードを指定します。自動判別させる事も可能ですが、このプロパティにて明示的に指定し判定ミスによる文字化け防止を図ることができます。

## 文字コード種別

自動判定	0
シフト JIS	1
日本語 EUC	2
ISO 2022 (JIS)	3
UTF-8	4
UTF-16 (Unicode)	5

---

**ScanLine プロパティ**

ビットマップを生成時に画像データの各行への添字によるアクセスを提供します。

**構文**

---

`expression.ScanLine[ Row ]`

`expression` 必ず指定します。**Canvas** インターフェースを表すオブジェクト式を指定します。

`Row` 整数型 (**Integer**) の値を使用します。0 基点でピクセル行を指定します。

## 解説

**BeginDoc** メソッドにてビットマップ出力を行うもの画面出力 (Window)、ファイル出力 (BMP, JPEG, PNG) を指定した時、**EndDoc** メソッド実行前に内部描画ビットマップへアクセス可能です。

データの一部にアクセスする時には利便性がありますが、データ全体をコピーする場合は **ClipboardCopy** メソッドを利用してクリップボード経由でアクセスした方が高速です。

---

## Paper プロパティ

**Paper** インターフェースへの参照を返し、用紙設定の取得・設定を行う機能を提供します。値の取得のみ可能です。

---

## Pen プロパティ

**Pen** インターフェースへの参照を返し、ペン (色・ブラシ) の取得・設定を行う機能を提供します。値の取得のみ可能です。

---

## Font プロパティ

**Font** インターフェースへの参照を返し、フォントの取得・設定を行う機能を提供します。値の取得のみ可能です。

---

## Paint プロパティ

**Paint** インターフェースへの参照を返し、中埋め、グラデーションなどの取得・設定を行う機能を提供します。値の取得のみ可能です。

---

### Shade プロパティ

**Shade** インターフェースへの参照を返し、影効果の取得・設定を行う機能を提供します。値の取得のみ可能です。

---

### Image プロパティ

**Image** インターフェースへの参照を返し、画像効果の取得・設定を行う機能を提供します。値の取得のみ可能です。

---

### Layers プロパティ

**Layers** インターフェースへの参照を返し、レイヤ・グループの取得・設定を行う機能を提供します。値の取得のみ可能です。

---

### Items プロパティ

**Items** インターフェースへの参照を返し、描画項目の取得・設定を行う機能を提供します。値の取得のみ可能です。

---

### Plugins プロパティ

**Plugins** インターフェースへの参照を返し、プラグインの取得・設定を行う機能を提供します。値の取得のみ可能です。

## Paper インターフェース

印刷用紙の情報の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、用紙の方向を変更します。

```
AutoTestObj.Canvas.Paper.Orientation := 1;
```

### 解説

詳細は描画命令ガイド を参照してください。

## PaperSize プロパティ

印刷用紙のサイズを用紙番号で返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

詳細は描画命令ガイド PLOT\_FORM コマンドを参照してください。

## Orientation プロパティ

印刷用紙の印刷方向を番号で返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

印刷用紙の印刷方向を番号で指定します。

ポートレート (縦書き)	1
ランドスケープ (横書き)	2

詳細は描画命令ガイド PLOT\_FORM コマンドを参照してください。

## PrintIndex プロパティ

印刷対象のプリンタ番号を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

パソコンにインストールされているプリンタのインデックス No で印刷対象のプリンタを変更する事が可能です。

詳細は描画命令ガイド PLOT\_FORM コマンドを参照してください。

## Pen インターフェース

中埋め、グラデーションの取得・設定を行うオブジェクトです。配列のように指定することで1~255までのペン情報を操作します。

### 使い方

次の使用例は、グラデーションの色設定を行います。

#### VisualBasic 6

```
Dim Col(4) As Integer
Dim Pos(4) As Double

...

` color 1,2,3,4
AutoTestObj.Canvas.Pen[1].Color = vbBlue
AutoTestObj.Canvas.Pen[2].Color = vbGreen
AutoTestObj.Canvas.Pen[3].Color = vbCyan
AutoTestObj.Canvas.Pen[4].Color = vbWhite
```

#### Delphi

```
var
  Col,Pos : PSafeArray;
  bxd,byd : array[0..0] of SAFEARRAYBOUND;
  px,py : Pointer;
  VarCol : array of Integer;
  VarPos : array of Double;

...

// color 1,2,3,4
AutoTestObj.Canvas.Pen[1].Color := clBlue;
AutoTestObj.Canvas.Pen[2].Color := clGreen;
AutoTestObj.Canvas.Pen[3].Color := clAqua;
AutoTestObj.Canvas.Pen[4].Color := clWhite;
```

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

## Color プロパティ

指定されたペンNoの色情報を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

色情報はシステムパレット・カラーの整数値で指定するか、青、緑、赤の RGB カラーの輝度で指定します。値 0x00FF0000 は最高の輝度、つまり純粋な青を示し、0x0000FF00 は純粋な緑を示し、0x000000FF は純粋な赤を示します。0x00000000 は黒、0x00FFFFFF は白です。

詳細は描画命令ガイド DEFINE\_PEN\_COLOR コマンドを参照してください。

### Width プロパティ

指定されたペンNoの線幅を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

#### 解説

線幅はピクセル単位で指定します。

詳細は描画命令ガイド DEFINE\_PEN\_WIDTH コマンドを参照してください。

### Style プロパティ

指定されたペンNoの線種を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

#### 解説

ペン線種 取得・設定値

実線	0
鎖線	1
点線	2
一点鎖線	3
二点鎖線	4
透明	5
塗りつぶし	6

詳細は描画命令ガイド DEFINE\_PEN\_TYPE コマンドを参照してください。

---

**Name プロパティ**

指定されたハッチ No の種別を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

詳細は描画命令ガイド DEFINE\_PEN\_COLOR コマンドを参照してください。

---

**BrushNo プロパティ**

指定されたハッチ No の種別を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

ハッチ種別 No およびコマンドの詳細は描画命令ガイド DEFINE\_HATCHING\_TYPE コマンドを参照してください。

## Font インターフェース

文字列の描画を行う際に使用するフォントの取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、フォントを変更します。

```
AutoTestObj.Canvas.Font.Name := 'MS Pゴシック';
```

### 解説

フォントを太字にしたり上付き文字にしたい場合は拡張書式を使用します。文字色の変更は現在のペン色を変更します。

詳細は描画命令ガイド を参照してください。

### Size プロパティ

使用中の Font のポイント数を返します。値の取得、設定が可能です。長整数型 (Long) の値を使用します。

### Hight プロパティ

使用中の Font のインチ数を返します。値の取得、設定が可能です。長整数型 (Long) の値を使用します。

### Name プロパティ

使用中の Font のフォント名を返します。値の取得、設定が可能です。文字列型 (String) の値

### CharSet プロパティ

使用中の Font のキャラクタセットを返します。値の取得、設定が可能です。長整数型 (Long) の値を使用します。

## 解説

キャラクタセットの詳細は描画命令ガイド [GRAPH\\_SYMBOL](#) コマンドの解説を参照してください。

## Paint インターフェース

中埋め、グラデーションの取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、グラデーションの色設定を行います。

#### VisualBasic 6

```

Dim Col(4) As Integer
Dim Pos(4) As Double

...

` color 1,2,3,4
AutoTestObj.Canvas.PenColor[1] = vbBlue
AutoTestObj.Canvas.PenColor[2] = vbGreen
AutoTestObj.Canvas.PenColor[3] = vbCyan
AutoTestObj.Canvas.PenColor[4] = vbWhite

AutoTestObj.Canvas.Paint.style := 1;
AutoTestObj.Canvas.Paint.Shape := 0;
AutoTestObj.Canvas.Paint.Strok := 1;
AutoTestObj.Canvas.Paint.Radial := 45;

AutoTestObj.Canvas.PenColor(0) = 1
AutoTestObj.Canvas.PenColor(1) = 2
AutoTestObj.Canvas.PenColor(2) = 3
AutoTestObj.Canvas.PenColor(3) = 4

AutoTestObj.Canvas.Paint.PenPos(0) = 0.0
AutoTestObj.Canvas.Paint.PenPos(1) = 20.0
AutoTestObj.Canvas.Paint.PenPos(2) = 40.0
AutoTestObj.Canvas.Paint.PenPos(3) = 100.0

```

#### Delphi

```

var
  Col,Pos : PSafeArray;
  bxd,byd : array[0..0] of SAFEARRAYBOUND;
  px,py : Pointer;
  VarCol : array of Integer;
  VarPos : array of Double;

...

// color 1,2,3,4
AutoTestObj.Canvas.PenColor[1] := clBlue;
AutoTestObj.Canvas.PenColor[2] := clGreen;
AutoTestObj.Canvas.PenColor[3] := clAqua;
AutoTestObj.Canvas.PenColor[4] := clWhite;

bxd[0].cElements := 4;
bxd[0].lLbound := 0;
Col := SafeArrayCreate(VT_I4, 1, bxd);
SafeArrayAccessData(Col, px);
VarCol := px;

```

```

byd[0].cElements := 4;
byd[0].lLbound := 0;
Pos := SafeArrayCreate(VT_R8, 1, byd);
SafeArrayAccessData(Pos, py);
VarPos:= py;

AutoTestObj.Canvas.Paint.style := 1;
AutoTestObj.Canvas.Paint.Shape := 0;
AutoTestObj.Canvas.Paint.Strok := 1;
AutoTestObj.Canvas.Paint.Radial := 45;

AutoTestObj.Canvas.Paint.PenColor[0] := 1;
AutoTestObj.Canvas.Paint.PenColor[1] := 2;
AutoTestObj.Canvas.Paint.PenColor[2] := 3;
AutoTestObj.Canvas.Paint.PenColor[3] := 4;

AutoTestObj.Canvas.Paint.PenPos[0] := 0.0;
AutoTestObj.Canvas.Paint.PenPos[1] := 20.0;
AutoTestObj.Canvas.Paint.PenPos[2] := 40.0;
AutoTestObj.Canvas.Paint.PenPos[3] := 100.0;

```

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

## PenColor プロパティ

グラデーションの描画において色の推移を制御する基準点の色(ペン No)を設定します。

### 構文

*expression*.Pen Color[*Index*] = *colorNo*

*expression* 必ず指定します。Fill インターフェースを表すオブジェクト式を指定します。

*Index* 整数型 (Integer) の値を使用します。基準点配列の引数を指定します。

*colorNo* 整数型 (Integer) の値を使用します。グラデーション基準点の色をペン No で指定します。

### 解説

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

## PenPos プロパティ

グラデーションの描画において色の推移を制御する基準点の位置を設定します。

**構文**

*expression*.Pen Pos[*Index*] = *Point*

*expression* 必ず指定します。Fill インターフェースを表すオブジェクト式を指定します。

*Index* 整数型 (Integer) の値を使用します。基準点配列の引数を指定します。

*Point* 実数型 (Double) の配列を使用します。グラデーション基準点の位置を百分率で指定します。

**解説**

グラデーションの基準点は必ず2点(始点:0%、終点:100%)を指定する必要があります。中間地点(例えば70%)で色を変更したい時は0%、70%、100%の3点を指定します。

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

**PenCount プロパティ**

グラデーションの描画において色の推移を制御する基準点の個数を取得・設定します。

**Style プロパティ**

図形の中埋め方法の種別を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

**解説**

図形の中埋め方法の種別

ハッチング	0
グラデーション (RGB)	1
グラデーション (HSV)	2

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

---

## Shape プロパティ

現在設定されているグラデーションの形状を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

グラデーションの形状

Line	0
Circle	1
Rect	2
Radio	3
Self	4
LabelNo	5

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

---

## Strok プロパティ

グラデーションの色を推移させる間隔(画面上の距離)を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

間隔の値はピクセル(Pixel)を指定します。この値は画面の拡大・縮小に追従しないため、拡大するほどグラデーションが細かくきれいになります。

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

---

## Radial プロパティ

グラデーション形状の回転角を返します。値の取得・設定が可能です。実数型 (Double) の値を使用します。

### 解説

グラデーション形状は **Shape** プロパティで指定します。

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

---

#### X, Y プロパティ

グラデーションの中心位置座標を返します。値の取得・設定が可能です。実数型 (**Double**)の値を使用します。

#### 解説

グラデーションを行う図形の中心座標を起点(0,0)として相対座標で指定します。

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

---

#### Width プロパティ

グラデーションを行う範囲を返します。値の取得・設定が可能です。実数型 (**Double**) の値を使用します。

#### 解説

**Width** プロパティで設定する範囲は **Shape** プロパティが Line の時は線の中央からの幅(線幅/2)、Circle の時は半径、Rectの時は図形の幅/2を指定します。

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

---

#### Sx, Sy プロパティ

グラデーション：焦点の座標を指定します。

パターン： 繰り返し時の軸方向 描画間隔を指定します。

実数型 (**Double**) の値を使用します。

#### 解説

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

## LblNo プロパティ

グラデーション、パターンで使用するラベル番号を指定します。整数型 (**Integer**) の値を使用します。

### 解説

詳細は描画命令ガイド EFACT\_FILL コマンドを参照してください。

## Shade インターフェース

影効果の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、開いているすべてのブックを閉じます。

#### VisualBasic 6

```

` color 1,2
AutoTestObj.Canvas.Pen[1].Color = vbBlue
AutoTestObj.Canvas.Pen[2].Color = vbGreen

AutoTestObj.Canvas.Shade.style = 1
AutoTestObj.Canvas.Shade.Strok = 10
AutoTestObj.Canvas.Shade.Side = 0
AutoTestObj.Canvas.Shade.Raial = 45
AutoTestObj.Canvas.Shade.StartColor = 1
AutoTestObj.Canvas.Shade.EndColor = 2

```

#### Delphi

```

// color 1,2
AutoTestObj.Canvas.Pen[1].Color := clBlue;
AutoTestObj.Canvas.Pen[2].Color := clGreen;

AutoTestObj.Canvas.Shade.style := 1;
AutoTestObj.Canvas.Shade.Strok := 10;
AutoTestObj.Canvas.Shade.Side := 0;
AutoTestObj.Canvas.Shade.Raial := 45;
AutoTestObj.Canvas.Shade.StartColor := 1;
AutoTestObj.Canvas.Shade.EndColor := 2;

```

詳細は描画命令ガイド EFACT\_SHADE コマンドを参照してください。

## Style プロパティ

現在設定されている影効果の種別を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

影効果の種別

なし	0
ノーマル (Normal)	1

ブロック(Block)	2
ベベル(Bevel)	3
陰影 (Shade)	4

詳細は描画命令ガイド EFACT\_SHADE コマンドを参照してください。

#### Strok プロパティ

影効果の幅を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

#### 解説

影効果の幅 (Normal: 影の長さ, Block: 影の描画位置、Bevel: 無効、Shade: 対象図形と影描画位置との距離) を指定します。

詳細は描画命令ガイド EFACT\_SHADE コマンドを参照してください。

#### Side プロパティ

影効果の表示位置 (内側、外側) を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

#### 解説

影効果の表示位置

外側	0
内側	1

詳細は描画命令ガイド EFACT\_SHADE コマンドを参照してください。

#### Radial プロパティ

影効果の伸びる方向を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

#### 解説

---

詳細は描画命令ガイド EFACT\_SHADE コマンドを参照してください。

---

#### StartColor, EndColor プロパティ

影効果の開始色、終了色のペン No を返します。値の取得・設定が可能です。長整数型 (**Long**) の値を使用します。

#### 解説

詳細は描画命令ガイド EFACT\_SHADE コマンドを参照してください。

## Image インターフェース

画像データの描画を行う際の画像効果の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、元画像にダークエンボス加工して描画します。

```
AutoTestObj.Canvas.Image.Style := 12;
AutoTestObj.Canvas.Draw(0, 0, "¥TES_TM.gif", 0, 0, 0, 0, 0);
```

### 解説

DelPlot で読み取れる画像ファイルを描画する際に指定されたエフェクト効果を加えます。

詳細は描画命令ガイド を参照してください。

## Style プロパティ

現在設定されている画像効果の種別を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

画像効果の種別

0	なし
1	XOR マスク
2	アンシェーブマスク
3	オーシャンブルー
4	ギザギザモザイク
5	グレイスケールカラー
6	セピアカラー
7	ネガポジ反転
8	拡散
9	二階調化(値 1:0-255(標準 127))
10	明るさ調整(値 1:(-255..0..255))
11	色合いの変更(値 1:元の色 BGR、 値 2:新しい色 BGR)

---

12	ダークエンボス
13	カラーエンボス
14	シャープ
15	ぼかし
16	サンパワー

詳細は描画命令ガイド EFACT\_IMAGE コマンドを参照してください。

---

#### Value, Value2 プロパティ

画像効果のパラメータ(値)を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

#### 解説

パラメータ(値を使用する画像効果については上記 Count プロパティ参照)

詳細は描画命令ガイド EFACT\_IMAGE コマンドを参照してください。

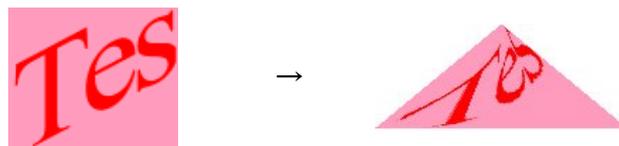
## Trans インターフェース

描画項目に座標変換を行い変形する際に使用する取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、画像を変形、描画します。

```
AutoTestObj.Canvas.Trans.Style := 1;  
AutoTestObj.Canvas.Trans.Width := 2;  
AutoTestObj.Canvas.Trans.Height := 2;  
AutoTestObj.Canvas.Trans.SourceX[0] := 0.0;  
AutoTestObj.Canvas.Trans.SourceY[0] := 0.0;  
AutoTestObj.Canvas.Trans.SourceX[1] := 1.0;  
AutoTestObj.Canvas.Trans.SourceY[1] := 0.0;  
AutoTestObj.Canvas.Trans.SourceX[2] := 0.0;  
AutoTestObj.Canvas.Trans.SourceY[2] := 1.0;  
AutoTestObj.Canvas.Trans.SourceX[3] := 1.0;  
AutoTestObj.Canvas.Trans.SourceY[3] := 1.0;  
AutoTestObj.Canvas.Trans.SourceX[0] := 0.5;  
AutoTestObj.Canvas.Trans.SourceY[0] := 0.5;  
AutoTestObj.Canvas.Trans.SourceX[1] := 1.0;  
AutoTestObj.Canvas.Trans.SourceY[1] := 0.0;  
AutoTestObj.Canvas.Trans.SourceX[2] := 0.0;  
AutoTestObj.Canvas.Trans.SourceY[2] := 1.0;  
AutoTestObj.Canvas.Trans.SourceX[3] := 2.0;  
AutoTestObj.Canvas.Trans.SourceY[3] := 1.0;  
AutoTestObj.Canvas.Draw(0, 0, "¥TES_TM.gif", 0, 0, 0, 0, 0);  
AutoTestObj.Canvas.Trans.Style := 0;
```



### 解説

変形前の座標と変形後の座標を与え、描画項目を変形できます。

詳細は描画命令ガイドを参照してください。

## Style プロパティ

変形の種別を指定します。

**解説**

変形の種別

0	なし
1	変形

詳細は描画命令ガイド DEFINE\_TRANS コマンドを参照してください。

**Width プロパティ**

変形対象に設定する基準格子の横軸数を指定します。

**解説**

詳細は描画命令ガイド DEFINE\_TRANS コマンドを参照してください。

**Height プロパティ**

変形対象に設定する基準格子の縦軸数を指定します。

**解説**

詳細は描画命令ガイド DEFINE\_TRANS コマンドを参照してください。

**SourceX,SourceY プロパティ**

変換前の図形に設定する格子軸の位置座標を%(パーセンテージ)で指定します。

**解説**

詳細は描画命令ガイド DEFINE\_TRANS コマンドを参照してください。

---

---

TargetX,TargetY プロパティ

変換前の格子点に対応する変形後座標要素の位置座標を%(パーセンテージ)で指定します。

**解説**

詳細は描画命令ガイド DEFINE\_TRANS コマンドを参照してください。

---

## Layers インターフェース

レイヤ・グループの取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、レイヤ・グループのスケールを変更します。

```
AutoTestObj.Canvas.Layers[0].zoom_type := 0;
```

### 解説

レイヤ・グループの設定を変更することでレイヤ内の描画項目の縮尺の変更や描画位置の移動、拡大などを制御します。

詳細は描画命令ガイド `DEFINE_LAYER` コマンド を参照してください。

---

## Index プロパティ

ページ内にあるレイヤ・グループ群の中で、そのレイヤ・グループの持つ Index を返します。値の取得が可能です。長整数型 (**Long**) の値を使用します。

---

## ID プロパティ

レイヤ・グループを指定時に設定された ID を返します。値の取得が可能です。長整数型 (**Long**) の値を使用します。

---

## Name プロパティ

レイヤ・グループを指定時に設定された ID を返します。値の取得が可能です。文字列 (**String**) の値を使用します。

## Zoom\_Type プロパティ

レイヤ・グループ内でのスケールを返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

## 解説

単位指定	値
フリースケール	0
ピクセル(Pixel)	1
インチ(Inch)	2
ミリメートル(mm)	3
経緯度(19座標系:メートル)	4
経緯度(UTM-GRS80)	5

描画命令や画面上からスケールの種別を変更すると、レイヤ・グループ内の拡大率(ExtendX, ExtendY)が変更されますが、現在のバージョンではこの設定を変更した場合、拡大率が同時に切り替わるものではありません。

詳細は描画命令ガイド DEFINE\_SCALE コマンドを参照してください。

## PositionX,PositionY プロパティ

レイヤ・グループ内の描画項目の描画位置をレイヤ単位で移動するオフセット値を返します。値の取得・設定が可能です。実数型 (Double) の値を使用します。

## 解説

詳細は描画命令ガイド DEFINE\_OFFSET コマンドを参照してください。

## ExtendX,ExtendY プロパティ

レイヤ・グループ内の描画項目をレイヤ単位で拡大する値を返します。値の取得・設定が可能です。実数型 (Double) の値を使用します。

## 解説

詳細は描画命令ガイド DEFINE\_SCALE コマンドを参照してください。

---

### Rotation プロパティ

レイヤ・グループ内の描画項目をレイヤ単位で回転する角度を返します。値の取得・設定が可能です。実数型 (Double) の値を使用します。

#### 解説

詳細は描画命令ガイド `DEFINE_ROTATE` コマンドを参照してください。

---

### RotBaseX, RotBaseY プロパティ

レイヤ・グループ内の描画項目をレイヤ単位で回転する回転中心位置を返します。値の取得・設定が可能です。実数型 (Double) の値を使用します。

#### 解説

詳細は描画命令ガイド `DEFINE_ROTATE` コマンドを参照してください。

---

## Items インターフェース

描画項目の設定情報の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、描画項目の種別を変更しています。

```
AutoTestObj.Canvas.Items[0].Value.comNo := 1;
```

### 解説

Canvas による描画命令によって登録されている描画項目の設定内容を取得・変更します。

詳細は描画命令ガイド を参照してください。

---

## Index プロパティ

描画項目の登録 Index を返します。値の取得が可能です。長整数型 (**Long**) の値を使用します。

---

## GroupID プロパティ

図形の描画項目で専用の色設定を持ったりなど、描画項目が関連を持つ項目を持つ場合、その項目の Index を返します。値の取得が可能です。長整数型 (**Long**) の値を使用します。

---

## Lbl プロパティ

描画項目のラベル名を返します。値の取得・設定が可能です。文字列 (**String**) の値を使用します。

---

## Txt プロパティ

描画項目が文字列を持つとき(テキスト描画命令など)文字列を返します。値の取得・設定が可能です。文字列 (**String**) の値を使用します。

---

### ItemArea プロパティ

描画項目の描画範囲を返します。値の取得が可能です。構造体を使用します。

Left	: Double	左端の座標
Top	: Double	上端の座標
Right	: Double	幅
Bottom	: Double	高さ

---

### Path プロパティ

描画項目が線分情報を持つ(ポリライン描画命令など)とき **DbPath** インターフェースへの参照を返し、線分情報の取得・設定を行う機能を提供します。値の取得のみが可能です。

---

### Mesh プロパティ

描画項目が立体形状情報を持つ(立体形状描画命令など)とき **DbMesh** インターフェースへの参照を返し、立体形状の取得・設定を行う機能を提供します。値の取得のみが可能です。

---

### Param プロパティ

描画項目が線分情報を持つときなど、座標点にパラメータ情報を持っていた時文字列を返します。値の取得・設定が可能です。文字列 (**String**) の値を使用します。

### 構文

*expression*.Param[ *Index* ]

*expression* 必ず指定します。Items インターフェースを表すオブジェクト式を指定します。

*Index* 整数型 (**Integer**) の値を使用します。Path など座標点の Index を指定します。

---

### Trans プロパティ

描画項目が変形情報を持つとき **Trans** インターフェースへの参照を返し、内容の取得・設定を行う機能を提供します。値の取得のみが可能です。

---

### Shade プロパティ

描画項目が影描画情報を持つとき **Shade** インターフェースへの参照を返し、内容の取得・設定を行う機能を提供します。値の取得のみが可能です。

---

### Paint プロパティ

描画項目が中埋め情報を持つとき **Paint** インターフェースへの参照を返し、内容の取得・設定を行う機能を提供します。値の取得のみが可能です。

---

### Image プロパティ

描画項目が画像効果情報を持つとき **Image** インターフェースへの参照を返し、内容の取得・設定を行う機能を提供します。値の取得のみが可能です。

---

### Layer プロパティ

描画項目がレイヤ・グループ情報を持つ(レイヤ・グループ)とき **Layer** インターフェースへの参照を返し、内容の取得・設定を行う機能を提供します。値の取得のみが可能です。

---

### Visible プロパティ

描画項目の表示・非表示の状態を返します。値の取得・設定が可能です。論理型 (**Boolean**) の値を使用します。

---



---

**Lock プロパティ**

描画項目の編集可能・不可の状態を返します。値の取得・設定が可能です。論理型 (Boolean) の値を使用します。

---

**Value プロパティ**

描画項目の設定を返します。すべての描画項目で使用されています。値の取得・設定が可能です。構造体を使用します。

pen	: Long	
fwd	: Long	
bak	: Long	
typ	: Long	
x, y, z	: Double	
comNo	: Long	描画項目の種別

---

**Value\_CI プロパティ**

描画項目の設定を返します。項目が DEFINE\_PEN\_COLOR の時、利用できます。値の取得・設定が可能です。構造体を使用します。

red	: Byte
green	: Byte
blue	: Byte
reserve	: Byte

---

**Value\_Ar プロパティ**

描画項目の設定をを返します。項目が図形項目の時、利用できます。値の取得・設定が可能です。構造体を使用します。

ra	: Double
rb	: Double
rot	: Double
sra	: Double
era	: Double

---

**Value\_Sy プロパティ**

描画項目の設定をを返します。項目が文字描画、画像描画などの時、利用できます。値の取得・設定が可能です。構造体を使用します。

h, w	: Double
rot	: Double
rw, rh	: Double
rx,ry,rz	: Double
typ	: Long
fcs	: Long
eff	: Long
lbl	: Long
fon	: Long
fon	: Long

---

**Value-Ta プロパティ**

描画項目の設定をを返します。項目が DEFINE\_TEXT\_ALIGN の時、利用できます。値の取得・設定が可能です。構造体を使用します。

ix, iy	: Long
jx, jy	: Long
swt	: Long

## Plugins インターフェース

プラグイン設定の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、プラグインの使用可能フラグを変更します。

```
AutoTestObj.Canvas.Plugins[0].Enable := True;
```

### 解説

DelPlot に登録されているプラグインの情報を閲覧、変更します。プラグイン固有のオプション、利用オプションの変更を行う機能ではありません。

詳細は描画命令ガイド を参照してください。

#### Enable プロパティ

プラグインの使用可能フラグを返します。DelPlot に登録できているプラグインで利用可能かどうかを示す値の取得・設定が可能です。論理型 (**Boolean**) の値を使用します。

#### Name プロパティ

プラグインの固有名を返します。値の取得が可能です。文字列 (**String**) の値を使用します。

#### Guid プロパティ

プラグインの GUID を返します。値の取得が可能です。文字列 (**String**) の値を使用します。

#### Style プロパティ

プラグインの種別を返します。値の取得が可能です。長整数型 (**Long**) の値を使用します。

## DbIPath インターフェース

線分情報の座標点の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、1つの座標点 (Index:0) 情報を変更します。

```
AutoTestObj.Canvas.Items.Path.PointX[0] := 0;  
AutoTestObj.Canvas.Items.Path.PointY[0] := 0;  
AutoTestObj.Canvas.Items.Path.Figure[0] := 0;
```

### 解説

線分情報 (点座標) の操作を提供します。

詳細は描画命令ガイド を参照してください。

## Count プロパティ

線分に登録された座標点数を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

## PointX,PointY プロパティ

線分の点座標の座標情報を返します。値の取得・設定が可能です。実数型 (Double) の一次元配列を使用します。

### 解説

詳細は描画命令ガイド PLOT\_PATHE コマンドを参照してください。

## Figure プロパティ

線分の点座標の属性情報を返します。値の取得・設定が可能です。長整数型 (Long) の一次元配列を使用します。

---

**解説**

LINE_TO	2
BEZIER_TO	4
MOVE_TO	6

線分を閉じるときは+1

詳細は描画命令ガイド PLOT\_PATH コマンドを参照してください。

## DbIMesh インターフェース

立体形状を構成する情報の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、立体形状の構成点を1つ登録し座標値を原点に設定します。

```
AutoTestObj.Canvas.Items.Mesh.Count := 1;  
AutoTestObj.Canvas.Items.Mesh.PointX[0] := 0;  
AutoTestObj.Canvas.Items.Mesh.PointY[0] := 0;  
AutoTestObj.Canvas.Items.Mesh.PointZ[0] := 0;
```

### 解説

立体形状を構成する座標情報の操作、平面情報への参照を提供します。

詳細は描画命令ガイド を参照してください。

### Count プロパティ

立体形状に登録された座標点数を返します。値の取得・設定が可能です。長整数型 (**Long**) の値を使用します。

### PlaneCount プロパティ

立体形状に登録された平面数を返します。値の取得・設定が可能です。長整数型 (**Long**) の値を使用します。

### PointX,PointY,PointZ プロパティ

立体形状の点座標の座標情報を返します。値の取得・設定が可能です。実数型 (**Double**) の一次元配列を使用します。

### 解説

詳細は描画命令ガイド PLOT\_MESH コマンドを参照してください。

## Plane プロパティ

**MeshPlane** インターフェースへの参照を返し、平面情報の取得・設定を行う機能を提供します。値の取得のみ可能です。

## MeshPlane インターフェース

立体形状の持つ平面情報の取得・設定を行うオブジェクトです。

### 使い方

次の使用例は、平面を1つ登録しその構成点数と座標 Index を設定します。

```
AutoTestObj.Canvas.Items.Mesh.PlaneCount := 1;  
AutoTestObj.Canvas.Items.Mesh.Plane[0].PosCount := 3;  
AutoTestObj.Canvas.Items.Mesh.Plane[0].PosIdx[0] := 0;  
AutoTestObj.Canvas.Items.Mesh.Plane[0].PosIdx[1] := 1;  
AutoTestObj.Canvas.Items.Mesh.Plane[0].PosIdx[2] := 2;
```

### 解説

DblMesh インターフェースに登録された座標点を繋いだ平面の情報を登録しています。内容を取得・設定します。

詳細は描画命令ガイド を参照してください。

### PosCount プロパティ

平面を構成する座標点の個数を返します。値の取得・設定が可能です。長整数型 (**Long**) の値を使用します。

### 解説

詳細は描画命令ガイド PLOT\_MESH コマンドを参照してください。

### PosIdx プロパティ

平面を構成する座標点の Index を返します。値の取得・設定が可能です。長整数型 (**Long**) の一次元配列を使用します。

### 解説

詳細は描画命令ガイド PLOT\_MESH コマンドを参照してください。

## DirTyp プロパティ

平面の方向を返します。値の取得・設定が可能です。長整数型 (Long) の値を使用します。

### 解説

詳細は描画命令ガイド PLOT\_MESH コマンドを参照してください。

## Execute インターフェース

DelPlot のウィンドウ上の各操作ツールを遠隔操作するオブジェクトです。**Canvas** インターフェースで描画したデータを活用する場合は **BeginDoc** メソッドの設定に注意が必要です。

### 使い方

次の使用例は、**Canvas** で描画した内容を印刷します。

```
AutoTestObj.Canvas.BeginDoc('window', true);
:
AutoTestObj.Canvas.LineTo(X, Y);
AutoTestObj.Canvas.EndDoc;
AutoTestObj.Execute.PrintOut;
```

指定しないと描画内容が保存されません

保存内容を印刷します

**BeginDoc** メソッド, **EndDoc** メソッドについての詳細はそれぞれの解説を参照してください。

## FileNew メソッド

データを新規作成します。

### 構文

*expression*.FileNew

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

## FileOpen メソッド

描画を行うためのデータファイルを指定するメソッドです。画面に描画結果を表示します。

### 構文

*expression*.FileOpen (*FileName*)

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

**FileName** 文字型 (**String**) の値を使用します。DelPlot で描画可能なフォーマット形式のファイルを指定します。

---

#### FileSave メソッド

描画結果をファイルに保存するメソッドです。フォーマット形式は指定したファイル名の拡張子で識別されます。

#### 構文

*expression*.FileSave (*FileName*)

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

**FileName** 文字型 (**String**) の値を使用します。DelPlot で保存可能なフォーマット形式のファイルを指定します。

---

#### FileSaveAs メソッド

描画結果をファイルに保存するメソッドです。フォーマット形式は指定したファイル名の拡張子で識別されます。

#### 構文

*expression*.FileSave (*FileName*, *MinPage*, *MaxPage*)

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

**FileName** 文字型 (**String**) の値を使用します。DelPlot で保存可能なフォーマット形式のファイルを指定します。

**MinPage** 整数型 (**Integer**) の値を指定します。保存する最初のページを指定します。

**MaxPage** 整数型 (**Integer**) の値を指定します。保存する最後のページを指定します。

#### 解説

引数 **MaxPage**にデータファイル内に存在しないページ数を指定した場合、存在する最大ページまでを出力の対象とします。

## AppExport メソッド

描画結果を外部アプリケーションに出力するメソッドです。現在出力可能な外部アプリケーションは MicrosoftExcel,Word、Justsystem 一太郎です。外部アプリケーションは実行環境にインストールされている必要があります。

### 構文

*expression*.AppExport (*Style*, *MinPage*, *MaxPage*, *SaveFileName*)

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

**Style** 整数型 (Integer) の値を指定します。外部アプリケーションに該当する番号を指定します。

MicrosoftWord	0
MicrosoftExcel	1
Justsystem 一太郎	2

**MinPage** 整数型 (Integer) の値を指定します。保存する最初のページを指定します。

**MaxPage** 整数型 (Integer) の値を指定します。保存する最後のページを指定します。

**SaveFileName** 文字型 (String) の値を使用します。Style で指定した形式のファイル名を指定します。

## BrushInport メソッド

ハッチング(中埋め)に活用するユーザーブラシ用のビットマップファイルを読み込みます。

### 構文

*expression*.BrushInport (*FileName*)

---

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

**FileName** 文字型 (**String**) の値を使用します。ビットマップファイル (.bmp) 形式のファイルを指定します。

## 解説

設定するユーザーブラシ、ビットマップ形式の詳細は描画命令ガイドの方を参照してください。

---

## PrintOut メソッド

描画結果をプリンタ印刷するメソッドです。実行時に表示される印刷ダイアログにて印刷ページ、プリンタ種別を選択できます。

## 構文

*expression*.PrintOut

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

## Undo メソッド

編集作業をひとつ前の状態に戻します。

## 構文

*expression*.Undo

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

## Redo メソッド

Undo を1つ取り消します。

## 構文

*expression*.Redo

---

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

### Copy メソッド

選択されたデータをクリップボードにコピーします。(エディットモード)

#### 構文

*expression.Copy*

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

#### 解説

データの選択には `SelectItem` メソッドを使用します。

---

### Cut メソッド

選択されたデータを切り取りクリップボードにコピーします。(エディットモード)

#### 構文

*expression.Cut*

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

#### 解説

データの選択には `SelectItem` メソッドを使用します。

---

### Paste メソッド

クリップボードの内容をデータに追加します。(エディットモード)

#### 構文

*expression.Paste*

---

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

### Delete メソッド

選択されたデータを削除します。(エディットモード)

#### 構文

*expression.Delete*

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

#### 解説

データの選択には **SelectItem** メソッドを使用します。

---

### Move メソッド

管理構造内の描画項目位置を移動します。(エディットモード)

#### 構文

*expression.Move( FromIndex, ToIndex )*

*expression* 必ず指定します。**Canvas** インターフェースを表すオブジェクト式を指定します。

**FromIndex** 長整数型 (Long) の値を使用します。移動する項目の Index を指定します。

**ToIndex** 長整数型 (Long) の値を使用します。移動先の Index を指定します。

---

### ClipboardCopy メソッド

描画結果をクリップボードへ出力するメソッドです。出力形式は BMP、WMF を選択可能です。

#### 構文

*expression.ClipboardCopy*

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

## 解説

出力形式は BMP、WMF の切り替えは **Canvas** インターフェースの **ClipBordForm** プロパティを使用します。詳細は **ClipBordForm** プロパティの解説を参照してください。

## SelectItem メソッド

1 ページ内でデータ(描画項目)を選択します。

## 構文

*expression*.**SelectItem** (*Index*, *Shift*)

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

**Index** 整数型 (**Integer**) の値を指定します。データリスト内の番号を指定します。

**Shift** 整数型 (**Integer**) の値を指定します。

1項目だけ選択	0
選択項目に追加	1

## PageFoward メソッド

描画対象のデータファイルが数ページに及ぶ時、現在ページの次のページを表示します。

## 構文

*expression*.**PageFoward**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

## PageBack メソッド

描画対象のデータファイルが数ページに及ぶ時、現在ページの前のページを表示します。

---

**構文**

*expression*.PageBack

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

**PageAdd メソッド**

現在編集中のデータにページを追加します。

**構文**

*expression*.PageAdd

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

**PageDelete メソッド**

現在編集中のデータから現在ページを削除します。

**構文**

*expression*.PageDelete

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

**PageRecheck メソッド**

データファイル内のページ数チェックをやり直します。

**構文**

*expression*.PageRecheck

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

## PageCheckStop メソッド

PagePecheck を停止します。

### 構文

*expression*.PageCheckStop

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

## AnimationStart メソッド

描画対象のデータファイルが数ページに及ぶ時、一定の間隔で切り替え表示を行いアニメーションのように表示します。

### 構文

*expression*.AnimationStart

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

### 解説

アニメーションの表示間隔、エンドレス指定は Canvas インターフェースの [AnimationTerm](#) プロパティ、[AnimationLoop](#) プロパティを使用します。詳細は [AnimationTerm](#) プロパティ、[AnimationLoop](#) プロパティの解説を参照してください。

---

## AnimationStop メソッド

アニメーションを終了させます。

### 構文

*expression*.AnimationStop

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

## Refresh メソッド

現在ページを再描画して表示をやり直します。

### 構文

*expression*.Refresh

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

## Group メソッド

選択中の項目をレイヤ・グループのメンバーにします。

### 構文

*expression*.Group

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

## UnGroup メソッド

選択中のレイヤ・グループを解除します。

### 構文

*expression*.UnGroup

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

## BringToFront メソッド

選択中のデータをページの最前列に移動します。

### 構文

---

***expression*.BringToFront**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

**SendToBack メソッド**

選択中のデータをページの最後尾に移動します。

**構文*****expression*.SendToBack**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

**MirrorX,MirrorY メソッド**

MirrorX 選択したデータを上下反転します。

MirrorY 選択したデータを左右反転します。

**構文*****expression*.MirrorX*****expression*.MirrorY**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

**RotLeft90,RotRight90 メソッド**

RotLeft90 選択したデータを左 90 度回転します。

RotRight90 選択したデータを右 90 度回転します。

**構文*****expression*.RotLeft90*****expression*.RotRight90**

---

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

#### LeftJustify,CenterJustify,RightJustify メソッド

LeftJustify 選択されたデータの最左端に位置合わせします。(左寄せ)

CenterJustify 選択されたデータの中央に位置合わせします。(中央寄せ)

RightJustify 選択されたデータの最右端に位置合わせします。(右寄せ)

#### 構文

*expression*.LeftJustify

*expression*.CenterJustify

*expression*.RightJustify

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

#### PointCut メソッド

線分編集にて選択点を切り取りクリップボードにコピーします。

#### 構文

*expression*.PointCut

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

#### PointCopy メソッド

線分編集にて選択点をクリップボードにコピーします。

#### 構文

*expression*.PointCopy

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

### PointPaste メソッド

線分編集にてクリップボードの選択点をデータに追加します。

#### 構文

*expression*.PointPaste

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

### PointDelete メソッド

線分編集にて選択点を削除します。

#### 構文

*expression*.PointDelete

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

### PointSelectAll メソッド

線分編集にて全点座標を選択状態にします。

#### 構文

*expression*.PointSelectAll

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---

### PointCancelAll メソッド

線分編集にて点座標の選択をすべて解除します。

#### 構文

---

*expression*.**PointCancelAll**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

### PointSelectReverse メソッド

線分編集にて点座標の選択状態を反転します。

#### 構文

*expression*.**PointSelectReverse**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

### PathOpen メソッド

線分の端点を開き、開いた線分(ポリライン)にします。

#### 構文

*expression*.**PathOpen**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---

### PathClose メソッド

線分の端点同士をつなぎ、閉じた線分(ポリゴン)にします。

#### 構文

*expression*.**PathClose**

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

---



---

**ChangeLinePoint メソッド**

線分編集にてコントロールポイントを持つ座標点(ベジェ曲線座標)からコントロールポイントを取り直線座標(角のある座標)にします。

**構文**

*expression*.ChangeLinePoint

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---



---

**ChangeCurvePoint メソッド**

直線座標(角のある座標)をベジェ曲線座標(コントロールポイントを持つ座標点)にします。

**構文**

*expression*.ChangeCurvePoint

*expression* 必ず指定します。Execute インターフェースを表すオブジェクト式を指定します。

---



---

**PointMode プロパティ**

線分編集にて点座標の編集モードの状態を取得・設定します。長整数型 (Long) の値を使用します。

全体編集モード	0
ポイント編集モード	1

---



---

**SelectMode プロパティ**

一括編集にて選択モード(項目の選択方法)の状態を取得・設定します。長整数型 (Long) の値を使用します。

全体(ラージ)選択モード	0
個別(マルチ)選択モード	1

## SelectWorks プロパティ

DelPlot の動作を切り替えます。長整数型 (Long) の値を使用します。

ビューアモード	0
エディットモード	1
コンバートモード	2

## EditAction プロパティ

エディットモードの編集状態を変更します。長整数型 (Long) の値を使用します。

一括編集機能	0
変形編集機能	1
図形編集機能	2
線分編集機能	3
文字編集機能	4
画像編集機能	5
3D 編集機能	6
データ挿入機能	7
(予約)	8
拡張ツール機能	9
パターン編集機能	10

## Scale プロパティ

表示の拡大率を百分率 (%) で指定します。長整数型 (Long) の値を使用します。

## SelectPoint メソッド

線形編集モードにて編集集中の線形の座標点を選択します。

## 構文

*expression*.SelectPoint(*Index*, *Shift*)

*expression* 必ず指定します。**Execute** インターフェースを表すオブジェクト式を指定します。

**Index** 整数型 (**Integer**) の値を指定します。線形データ内の番号を指定します。

**Shift** 整数型 (**Integer**) の値を指定します。

1項目だけ選択	0
選択項目に追加	1

#### ObjSelect プロパティ

3D 図形編集モードにて複数点を選択した場合の編集動作を指定します。長整数型 (**Long**) の値を使用します。

選択点移動	0
距離差移動	1
偏差移動	2

#### MeshEditMode プロパティ

3D 図形編集モードにて動作状態の取得・設定を行います。長整数型 (**Long**) の値を使用します。

カメラ 操作	0
基準点 操作	1
ポイント移動	2
拡大縮小	3
回転	4

#### MeshDispMode プロパティ

3D 図形編集モードにて画面表示方法を取得・設定を行います。長整数型 (**Long**) の値を使用します。

点	0
ワイヤー	1
面	2

---

**CtrlPointStyle プロパティ**

---

選択された座標(ベジェ曲線座標)のコントロールポイントの動作を切り替えます。長整数型 (Long) の値を使用します。

コントロールポイントなし(角のある座標)	0
座標点から等距離で連動する状態	1
連動する状態	2
非連動状態	3

## Menu インターフェース

DelPlot のウインドウに表示されているボタン・メニューの表示/非表示を制御します。

### 使い方

次の使用例は、「ファイルを開く」ボタンとメニュー項目を非表示にします。

```
AutoTestObj.Menu.FileOpen := false;
```

### File\_ プロパティ

メニューバーの「ファイル(F)」メニューの表示/非表示を制御します。

### FileNew プロパティ

「新規作成」ボタンとメニュー項目「ファイル(F)」 - 「新規作成」の表示/非表示を制御します。

### FileOpen プロパティ

「ファイルを開く」ボタンとメニュー項目「ファイル(F)」 - 「開く(O)」の表示/非表示を制御します。

### FileSave プロパティ

「ファイル保存」ボタンとメニュー項目「ファイル(F)」 - 「ファイル保存(S)」の表示/非表示を制御します。

---

**FileSaveAs プロパティ**

「名前を付けて保存」ボタンとメニュー項目「ファイル(F)」 - 「名前を付けて保存(E)」の表示/非表示を制御します。

---

**AppExport プロパティ**

メニュー項目「ファイル(F)」 - 「エクスポート(E)」の表示/非表示を制御します。

---

**FileInsert プロパティ**

メニュー項目「ファイル(F)」 - 「データ挿入(I)」の表示/非表示を制御します。

---

**BrushInport プロパティ**

メニュー項目「ファイル(F)」 - 「ブラシ読み込み(B)」の表示/非表示を制御します。

---

**PluginSetting プロパティ**

メニュー項目「ファイル(F)」 - 「プラグイン設定」の表示/非表示を制御します。

---

**Level0\_IO プロパティ**

メニュー項目「ヘルプ(H)」 - 「バージョン情報(A)」の表示/非表示を制御します。

---

**PrintSetting プロパティ**

メニュー項目「ファイル(F)」 - 「読捨データ I/O」の表示/非表示を制御します。

---

#### PrintOut プロパティ

「印刷」ボタンとメニュー項目「ファイル(F)」 - 「印刷(P)」の表示/非表示を制御します。

---

#### Edit\_ プロパティ

メニュー項目「編集(E)」メニューの表示/非表示を制御します。

---

#### Undo プロパティ

メニュー項目「編集(E)」 - 「元に戻す」の表示/非表示を制御します。

---

#### Redo プロパティ

メニュー項目「編集(E)」 - 「やり直し」の表示/非表示を制御します。

---

#### Copy プロパティ

「コピー」ボタンとメニュー項目「編集(E)」 - 「コピー」の表示/非表示を制御します。

---

#### Cut プロパティ

「切り取り」ボタンとメニュー項目「編集(E)」 - 「切り取り」の表示/非表示を制御します。

---

**Paste プロパティ**

「貼り付け」ボタンとメニュー項目「編集(E)」 - 「貼り付け」の表示/非表示を制御します。

---

**Delete プロパティ**

メニュー項目「編集(E)」 - 「削除」の表示/非表示を制御します。

---

**ClipboardCopy プロパティ**

メニュー項目「編集(E)」 - 「イメージのコピー(C)」の表示/非表示を制御します。

---

**View\_ プロパティ**

メニューバーの「表示(V)」メニューの表示/非表示を制御します。

---

**LogVieworSHow プロパティ**

メニュー項目「表示(V)」 - 「実行ログ表示(L)」の表示/非表示を制御します。

---

**Pallete プロパティ**

メニュー項目「表示(V)」 - 「パレット」の表示/非表示を制御します。

---

**ActionMode プロパティ**

メニュー項目「表示(V)」 - 「動作モード」の表示/非表示を制御します。

---

**ScaleChange プロパティ**

「拡大」「縮小」ボタンと拡大率プルダウンボックス、メニュー項目「表示(V)」 - 「拡大(Z)」の表示/非表示を制御します。

---

**PageFoward プロパティ**

「前ページ」ボタンとメニュー項目「表示(V)」 - 「前ページ(B)」の表示/非表示を制御します。

---

**PageBack プロパティ**

「次ページ」ボタンとメニュー項目「表示(V)」 - 「次ページ(N)」の表示/非表示を制御します。

---

**AnimationStart プロパティ**

「アニメーション再生」ボタンとメニュー項目「表示(V)」 - 「再生(P)」の表示/非表示を制御します。

---

**AnimationStop プロパティ**

「アニメーション停止」ツールボタンとメニュー項目「表示(V)」 - 「停止(S)」の表示/非表示を制御します。

---

**SystemOption プロパティ**

メニュー項目「表示(V)」 - 「オプション(O)」の表示/非表示を制御します。

---

**Refresh プロパティ**

「再表示」ツールボタンとメニュー項目「表示(V)」 - 「再表示」の表示/非表示を制御します。

---

**Help\_ プロパティ**

メニューバーの「ヘルプ(H)」メニューの表示/非表示を制御します。

---

**ExtRegisterSet プロパティ**

メニュー項目「ヘルプ(H)」 - 「拡張子\*.plt を DelPlot に関連付ける」の表示/非表示を制御します。

---

**RegistryClear プロパティ**

メニュー項目「ヘルプ(H)」 - 「レジストリのクリア」の表示/非表示を制御します。

---

**About プロパティ**

メニュー項目「ヘルプ(H)」 - 「バージョン情報(A)」の表示/非表示を制御します。

---

**PageControl プロパティ**

ステータスバー上、ページ移動・スライダー の表示/非表示を制御します。

---

**PageRecheck プロパティ**

ステータスバー上、「ページ再調査」ボタンの表示/非表示を制御します。

---

PageCheckStop プロパティ

ステータスバー上、「ページ調査中止」ボタンの表示/非表示を制御します。

---

PageAdd プロパティ

ステータスバー上、「ページ追加」ボタンの表示/非表示を制御します。

---

PageDel プロパティ

ステータスバー上、「ページ削除」ボタンの表示/非表示を制御します。

## DelPlot ver.1.9.0 DCOM ガイド

---

2012/04/01. 第1版発行

Copyright ©2003-2012 Table Soft.

発行 テーブルソフト  
Email : tes@hello.jp  
http : //www.hello.ne.jp/tes/

発行者 平田 治

### ご注意

- ・ 本製品の一部または全部を弊社の許可無く複写・複製することは禁じます。
- ・ 本製品の内容・仕様は改正・改善のため予告なく変更することがあります。
- ・ 本マニュアル記載のソフトウェア使用許諾書に基づいて個人で使用する場合を除いて、弊社の許諾なしにこのソフトウェアおよびマニュアルを使用することを固くお断りします。
- ・ このソフトウェアおよびマニュアルを運用した結果の影響については責任を負いかねますので、ご了承ください。
  - ・ 本製品の内容につきましては万全を期しておりますが、万一ご不審な点やお気づきの点がございましたら、弊社までご連絡ください。