
mitsubishi

三菱Web地理情報システム構築パッケージ



PreServ for Web Ver.5

MapViewer

Q&A 集

ver.2

はじめに

本ドキュメントは、PreSerV for Web MapViewer について問い合わせのあった件について Q&A 形式にまとめたものです。

目次

1	技術的な質問	1
1.1	図形の固定表示について	1
1.2	表示優先度について	2
1.3	描画プライオリティが同一の場合の描画順について	3
1.4	イメージシンボルの表示方法について	4
1.5	イメージシンボルのファイルフォーマットについて	5
1.6	ユーザー情報のデータベース登録の有無について	6
1.7	クローン時のユーザ情報について	7
1.8	文字の全角 / 半角について	8
1.9	文字図形オブジェクトの文字列変更について	9
1.10	塗潰し属性の透過について	11
1.11	塗りつぶしのみの透過について	12
1.12	スクリーンの図形直接描画について	13
1.13	スクリーンに直接描画する図形色について	14
1.14	スクリーンに直接描画する図形の線幅について	15
1.15	ポリゴンの始点終点座標について	16
1.16	一筆書きでないポリゴン描画について	17
1.17	psvScreen#get_scale()メソッドについて	18
1.18	PreSerV のフォントについて	19
1.19	重量、軽量コンポーネントについて	20
1.20	psvTree の readExternal について	21
1.21	圧縮転送の性能について	22
1.22	psvJavaScreen#init で Exception について	23
1.23	地図の重ね合わせについて	24
1.24	地図のイメージ出力について	25
1.25	スケール値 (解像度) について	26
1.26	スケール値 (アスペクト比) について	27
1.27	地図表示範囲 (最小 / 最大) について	29
1.28	イメージシンボルについて	30
1.29	イメージシンボルについて 2	31
1.30	DM 地図で線が表示されない現象について	33
1.31	DM 地図表示時のユーザー線種エラーについて	34
1.32	Raster 地図使用時のバージョン変更で発生する、クラス未取得エラーについて	35

1.33	図形読み込み失敗の原因調査方法について	36
1.34	レイヤー番号未設定の psvLayerAttrib 設定について.....	38
1.35	Raster 地図表示で OutOfMemory エラーが発生してしまう現象について	39
1.36	psvMdmMapEx から psvLayer が取得できない現象について	40
1.37	印刷時の ScreenDraw メソッドが複数回呼び出される.....	41
1.38	レイヤを使った表示切替について	42
1.39	J a v a 2 を利用した点線描画について	43
1.40	表示画面が反転表示される	44
1.41	地図の回転について.....	45
1.42	psvBinaryInfo (Ver5 バイナリ特有属性情報) について.....	46
1.43	透過率を指定した直接描画について	48
1.44	レイヤの縮尺による表示切替について	49
1.45	レイヤの設定可能属性について	51
1.46	同一座標の図形が複数ある場合の優先順位について	52
1.47	メッシュの検索について.....	53
1.48	psvVectorMap#search_nearest()メソッドについて	54
1.49	レイヤ登録数について	56
1.50	マップ登録数について	57
1.51	psvJavaScreen#get_wc_width()メソッドが返す WC 値について	58
2	その他の質問.....	59
2.1	PreSerV for Web 署名について	59
2.2	GIF ライセンスについて	60

1 技術的な質問

技術的な質問についての Q&A です。

1.1 図形の固定表示について

Ver.3

Ver.4

Ver.5

Q 地図の縮尺が異なった場合でも、図形の大きさを固定にしたいのですが出来るのでしょうか？

A 個々の図形に対して大きさを固定にすることは出来ませんが、レイヤ単位で図形の大きさを固定にすることが出来ます。（固定表示できる図形は文字、シンボル、イメージシンボルのみです）
psvLayer#set_obj_range()メソッドを利用することで、レイヤ単位で図形の表示サイズを設定することが出来ます。

リスト1 コーディング例（表示サイズの指定）

```
#shape 図形オブジェクト
#layer 地図オブジェクトに登録している psvLayer オブジェクト

//レイヤ番号 1 を指定し、レイヤオブジェクトに登録
shape.layer(1,layer);

//レイヤ番号 1 を持つ図形オブジェクトの表示上下限値を指定します。
//上下限値に 20 を指定（DC 座標値）
layer.set_obj_range(1,20,20);
```

1.2 表示優先度について

Ver.3

Ver.4

Ver.5

- Q 図形の画面表示優先度はどのように指定するのでしょうか？
- A 図形オブジェクトの属性 `psvShape#attrib()`メソッドで描画優先度 (1 ~ 1 0 2 4) を指定することができます。5 ビット目以降が描画優先度になります。
(PreSerV API ドキュメント、オブジェクト属性を参照)

リスト2 コーディング例 (表示優先度属性の指定)

```
#shape 図形オブジェクト
//図形オブジェクト属性の設定
//可視性、検知性、表示優先度 4 を設定
shape.attrib(psvAttrib.VISIBLE| psvAttrib.DETECTIVE|(4<<5));
```

またレイヤ単位で表示優先度を指定することも出来ます。優先度の指定は `psvLayer#attrib()`メソッドで行います。指定方法は図形オブジェクトと同様です。

リスト3 コーディング例 (レイヤ表示優先度属性の指定)

```
#layer レイヤオブジェクト
//レイヤオブジェクト属性の設定
//レイヤ番号 1 を持つ図形オブジェクトに可視性、表示優先度 4 を設定
layer.attrib(1,psvAttrib.VISIBLE |(4<<5));
```

1.3 描画プライオリティが同一の場合の描画順について

Ver.3

Ver.4

Ver.5

- Q** オブジェクトの属性として描画のプライオリティがありますが、プライオリティが同一の場合にどのように描画されるのでしょうか？
- A** 同一プライオリティでの描画順は、メッシュ情報(psvTree)に登録されている図形情報を取得した順番で描画されております。psvTree から取得できる図形の順番を指定することはシステム上行うことが出来ません。

1.4 イメージシンボルの表示方法について

Ver.3

Ver.4

Ver.5

- Q** イメージシンボルを表示したいのですが、どうすれば使用できるのでしょうか？
- A** イメージ管理クラス `psvImageManager` でイメージの読み込み、管理を行います。読み込みは URL か String でイメージファイルパスを指定するかファイル名を記述することで行えます。
読み込んだイメージは `psvImageSymbol` コンストラクタにイメージ管理クラスに登録したキーを指定することで使用することが出来ます。

リスト4 コーディング例(イメージシンボルの作成)

```
//イメージ管理クラスの作成
psvImageManager imgmng = new psvImageManager();
//イメージ管理クラスにイメージパス、キーを指定でイメージを読み込む
//カレントディレクトリにある melco.gif ファイルを読み込む
imgmng.addImage("melco.gif","melco");
//イメージ管理クラスを psvImageSymbol クラスに登録
psvImageSymbol.setManager(imgmng);

//イメージシンボルオブジェクト作成
//イメージ管理クラスに登録したキーを指定しイメージシンボルオブジェクトを作成
psvShape any = new
psvImageSymbol((Object)"melco",(short)0,10,40,80,20,dc,null);
```


1.5 イメージシンボルのファイルフォーマットについて

Ver.3

Ver.4

Ver.5

- Q** イメージシンボルのフォーマットとして、PNGは使用可能でしょうか。マニュアルはGIF、JPGとなっていますが、実際はPNGを指定しても表示可能のようですが？
- A** Java1.3以降を使用する場合は、Javaがjpg,gifと同じようにpngが使えることになっているため、使用可能です。

1.6 ユーザー情報のデータベース登録の有無について

Ver.3

Ver.4

Ver.5

- Q** psvShape#set_value メソッドでユーザ情報を定義し、データベースに登録する際に、ユーザ情報を登録する / しないを定義できるのでしょうか？
- A** 図形登録形式をシリアルライズ形式で保存する場合、psvArgs#setSerial()メソッドのパラメータに true (シリアルライズを行う)、false (シリアルライズを行わない) を設定することで、ユーザ情報の書き込みの有無を指定できます。データベースに限らず、ファイルに出力などストリームに乗せる処理も同様となります。

1.7 クローン時のユーザ情報について

Ver.3

Ver.4

Ver.5

- Q** psvShape#clone()メソッドでオブジェクトのコピーを作成した場合、ユーザ情報もコピーされるのでしょうか？
- A** クローン時にはユーザ情報もコピーされ情報は保持されます。ただし value 値に入れたオブジェクトが clone をサポートしていない場合以下のようにになります。

```
/*-----JAVADOC (Cloneable)抜粋
Cloneable インタフェースを実装しないインスタンスの複製 (コピー)
を試みると、結果として例外 CloneNotSupportedException がスローされ
ます。
-----*/
```

また psvMdmMapEx を利用してデータベースから図形オブジェクトを読み込んだ図形をクローンした場合、ユーザ情報に MDM キーが登録されています。クローンした図形をデータベースに登録する場合、ユーザ情報から MDM キーを削除してから登録を行う必要があります。

リスト5 コーディング例 (ユーザ情報から MDM キーの削除)

```
# any = 図形オブジェクト

//MDM キーは"mdm_key"をキーとしてユーザ情報に登録されている
if(any.get_values("mdm_key") != null){
    //キーを指定し MDM キーの削除
    any.get_context().get_args().del("mdm_key");
}
```

1.8 文字の全角 / 半角について

Ver.4

Ver.5

- Q** 文字図形オブジェクトを半角文字で生成し、図形オブジェクトから文字列を取得すると、全角文字が返ってきます。PreSerV の仕様としては、半角は全角に変換されるのでしょうか？
- A** PreSerV で半角文字列を扱う場合には、文字オブジェクト `psvFontData` に文字オブジェクトの属性 `psvTextContext` の半角対応文字種別 (`psvTextContext#TX_VFONT2`) を指定していただくことで、半角文字を利用することが出来ます。全角文字が取得される場合は、文字種別に `psvTextContext#TX_VFONT` が指定されていることが考えられます。

1.9 文字図形オブジェクトの文字列変更について

Ver.3

Ver.4

Ver.5

- Q 文字図形オブジェクトの文字列を変更する方法として、下記方法で問題ないでしょうか？

リスト6 コーディング例(変更方法)

```
例) A A A => B B Bに変更
// 図形の文字情報の設定
shapeObj.get_context().get_tx_attrib().tx_code =
psvFontData.convert(nameStr);
```

- A 文字列の属性や外接長方形が変化しない文字図形オブジェクト（文字列長、回転角が同じ）の場合は、文字列の変更が可能です。上記コーディングは前述で述べた条件の場合でのみ有効となります。上記条件以外の場合は、文字列の変更ではなく、変更元の図形オブジェクトを削除してから新しい、文字図形オブジェクトを作成/登録する必要があります。これは、コンストラクタから文字図形オブジェクトを作成するタイミングで図形の外接長方形を算出しており、図形管理を行うオブジェクト psvTree クラスには、このコンストラクタで算出した領域で図形を管理する必要があるためです。PreSerV では文字種別が TX_VFONT の場合、EUC コードのバイト配列、その他の文字種別は String で保持します。以下に文字列の取得方法と、変更が可能な場合の文字情報の設定例を記述します。

リスト7 コーディング例(文字列取得例)

```
# font = psvFontData
String moji = null;
if(shapeObj.get_context().get_tx_attrib().tx_type
    ==psvTextContext.TX_VFONT){
    // EUC コードバイト配列から SJIS コードバイト配列に変換し
    文字列として取得
    moji = new String(psvFontData.reverse(
        (byte[])font.get_context().get_tx_attrib().tx_code));
}else{ // psvTextContext.TX_VFONT 以外の文字種別
    // 図形の文字情報の取得
    moji = (String)font.get_context().get_tx_attrib().tx_code;
}
```

リスト8 コーディング例(文字情報の設定例)

```
例) A A A => B B Bに変更
if(shapeObj.get_context().get_tx_attrib().tx_type
    ==psvTextContext.TX_VFONT){
    // 図形の文字情報の設定
    // 文字列を EUC コードバイト配列に変換し設定
    shapeObj.get_context().get_tx_attrib().tx_code =
    psvFontData.convert(nameStr);
}else{ // psvTextContext.TX_VFONT 以外の文字種別
    // 図形の文字情報の設定
    shapeObj.get_context().get_tx_attrib().tx_code = nameStr;
}
```

1.10 塗り潰し属性の透過について

Ver.3

Ver.4

Ver.5

- Q 図形を塗り潰し透過で表示したいのですが、どうすればできるのでしょうか？
- A 図形の塗り潰し透過を行う場合は `jp.co.melco.preserv.j2` パッケージに含まれる `psvJavaScreen2D` クラスを使用します。
図形の塗り潰し透過は、図形塗り潰し属性の 16 ビット目にフラグを設定し、アルファ値を 8 ビット目に設定します。

リスト9 コーディング例(塗り潰し透過)

```
//スクリーンクラスの作成
screen = new psvJavaScreen2D(canvas, 0, 0, 1000, 1000);

//ポリゴン点配列作成
psvPoint[] ps = new psvPoint[3];
ps[0] = new psvPoint(x-100,y-100);
ps[1] = new psvPoint(x+100,y-100);
ps[2] = new psvPoint(x-100,y+100);
//図形属性設定
psvDrawContext context = new psvDrawContext();
//表示属性
context.attrib(psvAttrib.VISIBLE);
//塗り潰し属性にアルファ値、透過フラグを設定
context.paint_attrib(100 << 8 | 1 << 16);
context.set_color(new psvColor(new Color(0,0,255)));
psvPolygonData polygon = new psvPolygonData(ps,context,null);
```

1.11 塗りつぶしのみの透過について

Ver.3

Ver.4

Ver.5

- Q** PreSerV で透過表示を行うと境界線も透過になってしまいます。枠線は透過とせず、塗りだけ透過にする方法はないでしょうか？
- A** PreSerV では、透過の指定を行うと図形オブジェクトの色に対して透過色を適用するため、指定した色が全て透過になってしまいます。しかし、Java2 環境で一時的な描画や、色の再設定を行うことにより、個々（線色、塗り潰し色）に対して透過の指定を行うことが出来ます。一時的な理由として、JDK1.1 との互換のためデータベース等に保存する場合にシリアライズを行うとこのアルファ値が保存されないためです。指定方法は、図形オブジェクトの透過フラグを設定せず、通常の Java の色の指定にアルファ値を設定します。

リスト10 コーディング例（個々の塗り潰し透過指定）

```
//ポリゴン点配列作成
psvPoint[] ps = new psvPoint[3];
ps[0] = new psvPoint(x-100,y-100);
ps[1] = new psvPoint(x+100,y-100);
ps[2] = new psvPoint(x-100,y+100);
//図形属性設定
psvDrawContext context = new psvDrawContext();
//表示属性
context.attrib(psvAttrib.VISIBLE);
//塗り潰し属性に不透明を指定 透過フラグ OFF で描画
context.paint_attrib(2);
//図形色を設定（線色、塗り潰し色を個々に設定）
context.set_color(new psvColor(new Color(0,0,255,255),
                                new Color(0,0,255,100)));
psvPolygonData polygon = new psvPolygonData(ps,context,null);
```


1.12 スクリーンの図形直接描画について

Ver.3

Ver.4

Ver.5

- Q** psvJavaScreen#rectangle()メソッドを使用し矩形を描画しようとしたのですが、矩形が描画されません。スクリーンに用意されている図形描画メソッドを使用する場合、必要な設定があるのでしょうか？
- A** スクリーンに直接描画を行うような場合は、一度オフスクリーンを解除してから操作してください。オフスクリーンの切り替えはpsvScreen#set_offscreen_mode()メソッドで行います。

リスト11 コーディング例 (オフスクリーンの使用)

```
screen = new psvJavaScreen2D(canvas, 0, 0, 100, 100);
screen.set_color(new psvColor(Color.red,Color.blue));
// オフスクリーン解除
screen.set_offscreen_mode(false);
//矩形描画
screen.rectangle(1,1,10,10);
// オフスクリーン設定
screen.set_offscreen_mode(true);
```

1.13 スクリーンに直接描画する図形色について

Ver.3

Ver.4

Ver.5

Q スクリーンに直接描画する図形に色をつけるにはどうすればよいのですか？

A psvJavaScreen#set_color()メソッドで行います。色を設定後、描画メソッドを呼び出してください。

リスト12 コーディング例(色指定)

```
//線色、塗り潰し色指定
screen.set_color(new psvColor(Color.red,Color.blue));
//矩形描画
screen.rectangle(1,1,10,10);
```

1.14 スクリーンに直接描画する図形の線幅について

Ver.3

Ver.4

Ver.5

- Q** スクリーンに直接描画する図形の線幅を指定するにはどうすればよいのですか？
- A** psvJavaScreen#set_line_width()メソッドで行います。線幅を設定後、描画メソッドを呼び出してください。

リスト13 コーディング例 (線幅指定)

```
//線幅 2 を指定
screen.set_line_width(2);
//矩形描画
screen.rectangle(1,1,10,10);
```

1.15 ポリゴンの始点終点座標について

Ver.3

Ver.4

Ver.5

- Q** ポリゴン図形描画の際，図形頂点の始点，終点は同一座標でなければならないのでしょうか？
- A** PreSerV のポリゴン座標についてですが、始点と終点は、同一座標で作成する必要があります。ただし、psvPolygonData のコンストラクタから指定した場合は、PreSerV 内で始点と終点が一致していない場合は、自動的に 1 点追加されます。

1.16 一筆書きでないポリゴン描画について

Ver.3

Ver.4

Ver.5

Q 穴あきドーナツイメージのような一筆書きでないポリゴンを書くことはできるのでしょうか？

A 申し訳ございませんが、現在の PreSerV の機能では、「穴あきドーナツイメージのような図形」を表現できる機能はサポートしておりません。
中抜きポリゴン(穴あきドーナツイメージのような図形)を表現するためには、psvPolygonData クラスを継承して中抜きポリゴンを一筆書きのポリゴンに変換して描画するクラスを別途作成する必要があります。

1.17 psvScreen#get_scale()メソッドについて

Ver.3

Ver.4

Ver.5

- Q** psvScreen#get_scale()メソッドの戻り値は、どのような値なのでしょうか。psvJavaScreen#get_view()メソッドのX軸方向やY軸方向の差を計算した値とは違う値になっています。直交座標系なので、get_view()メソッドの差分から、スクリーンの表示している範囲の距離が取得できると考えておりました。レイヤの表示範囲の設定(set_range)は、このget_scale()メソッドの値で設定されているようですが、get_scale()メソッドとget_view()メソッドではどのような違いがあるのでしょうか？
- A** get_scale()メソッドの戻り値はスケール値です。縮尺値 $1 / XXX$ の XXX を int で返します。
get_view()メソッドは現在表示中の左下 WC 座標と右上 WC 座標を返します。

1.18 PreSerV のフォントについて

Ver.3

Ver.4

Ver.5

- Q** PreSerV のフォントとして持っている、明朝・ゴシック・ベクトルについて、明朝フォント / ゴシックフォントは OS (ブラウザ) が持っているフォントを使用していると思うのですが、Windows 系の OS の場合、実態として使用されているフォントは何になりますか？
- A** PreSerV では、フォント指定をフォント論理名として明朝の場合 "serif" ゴシックの場合 "sansserif" を指定しております。
J R E 等をご利用の場合は、
J R E インストールディレクトリ . . . ¥lib¥font.properties.java
で定義されたフォントのうち Windows のデフォルトでは serif=M S 明朝
sansserif=M S ゴシックが定義されており、OS によって指定がちがうよう
です。I E では内部で値を固定に持っているようです。

1.19 重量、軽量コンポーネントについて

Ver.3

Ver.4

Ver.5

Q psvJCanvas クラスが重量コンポーネント `java.awt.Canvas` を継承しているため、メニューのポップアップ等、軽量コンポーネントが裏に隠れてしまいます。`jp.co.melco.preserv.j2` パッケージにある `psvJComponent` クラスを `psvJCanvas` クラスの代わりに使用しようと思うのですが、使い方は `psvJCanvas` クラスと同じでしょうか？

A `psvJComponent` は、軽量コンポーネントに対応するためのものをご利用方法として `psvJCanvas` を `psvJComponent` に切り替えることで問題ありません。利用法につきましても `psvJCanvas` と同じになります。

ただし、コンポーネントから `Graphics` オブジェクトが取得出来ない状態ではスクリーンクラスに登録することが出来ません。

`Graphics` オブジェクトが取得できない例としてアプレットの `init` 時に `psvJComponent` のインスタンスを作成した状態や登録するコンポーネント (`Frame`) が非表示の場合が考えられます。

1.20 psvTree の readExternal について

Ver.3

Ver.4

Ver.5

- Q** psvTree の writeExternal()メソッドを使ってファイルに書き出した内容を readExternal()で復元しました。
結果として、psvTree に含まれている図形オブジェクトはきちんと復元されているのですが、psvLayerSelect での検索で、該当するはずの図形が検索されなくなりました。psvTree にふくまれる図形オブジェクトには、レイヤ番号は設定されているのですが、何故でしょうか？
- A** 図形オブジェクトで管理されているのは、レイヤ番号のみでレイヤ情報のシリアライズは、行っておりません。今回のような場合は、復元した psvTree に含まれる図形オブジェクトを取得し、レイヤ情報を再設定する必要があります。図形オブジェクトの取得は、psvTree に対し検索処理を行うことで取得することが出来ます。

リスト14 コーディング例(レイヤ情報の再設定)

```
//psvTree の復元
psvTree tree = new psvTree();
tree.readExternal(new ObjectInputStream(new FileInputStream("ファイル")));
psvRegionBounds rg = tree.getBounds();

// psvTree の領域で検索
psvSelect rs = new psvRegionSelect(null, rg);
psvSearchable psch = new psvSearch(rs, false);
if(tree.search(psch)){
    Enumeration objs = psch.elements(0);
    while(objs.hasMoreElements()){
        // 検索結果の図形オブジェクト取得
        psvShape shape = (psvShape)objs.nextElement();
        /* レイヤ情報の再設定
         * _ltable は、psvLayer オブジェクト
         */
        shape.layer(shape.layer(), _ltable);
    }
}
```

1.21 圧縮転送の性能について

Ver.3

Ver.4

Ver.5

Q ドキュメントに記載されている方法に沿って地図データの圧縮転送を試みたのですが、通信性能面での効果を見出すことができませんでした。結果から判断すると、本当に地図データが圧縮されているのかどうか、良くわかりません。

LAN 環境では、効果が出ないものなのでしょうか？

A PRESERV の圧縮転送は JAVA の通常クラス `java.util.zip.outputstream` を利用しています。

事前に圧縮したデータをサーバに置いておき、`psvCompressTransfer` でデータを取得することも可能です。その条件で試してみれば、確実に圧縮時の転送時間が測定できます。

1.22 psvJavaScreen#init で Exception について

Ver.3

Ver.4

Ver.5

Q psvJavaScreen の init で Exception が起きてしまいます。

コンストラクタは以下のような感じです。

```
new psvJavaScreen(canvas, 0, 0, 100, 100)
```

変数 canvas は psvJCanvas の継承クラスで null ではありません。

```
java.lang.NullPointerException
```

```
at jp.co.melco.preserv.psvJavaScreen.dc_resize(psvJavaScreen.java)
```

```
at jp.co.melco.preserv.psvJavaScreen.init(psvJavaScreen.java)
```

```
at jp.co.melco.preserv.psvJavaScreen.<init>(psvJavaScreen.java)
```

考えられる原因は何でしょうか？

A イメージの作成に必要な Graphics オブジェクトがまだ作成されていない状態で psvJavaScreen の作成を行ったために発生したエラーとなっております。

Frame オブジェクトに登録した psvJCanvas オブジェクトから Graphics オブジェクトが取得できる状態をご利用いただけますようお願いいたします。

基本的には、psvJCanvas が画面に表示された状態でしたら Graphics オブジェクトが取得できる状態となっております。

1.23 地図の重ね合わせについて

Ver.3

Ver.4

Ver.5

- Q** 何枚かの地図を重ねて表示し、その地図を少しずらして見たいのですが、そのようなことは可能でしょうか？
- A** 2つのスクリーンオブジェクトを作成して、キャンバスオブジェクトに登録します。1つの地図だけをずらす場合には、片方のスクリーンオブジェクトのみ view 等の操作を行います。

1.24 地図のイメージ出力について

Ver.3

Ver.4

Ver.5

- Q** PreSerV で表示された地図を、イメージファイル(jpeg 等)に落とす機能はありますか？
- A** PreSerV にはありませんが、キャンバスの表示内容を java.awt.image 形式にて取り出すことが可能です。
java.awt.image から JPEG などのイメージ形式に変換する JAVA クラスはフリーのものを含め各種ありますので、それをご利用ください。

1.25 スケール値（解像度）について

Ver.3

Ver.4

Ver.5

Q PreSerV におけるスケール値は、どのように算出されているのでしょうか？

A (1)ディスプレイの場合

Java のメソッド (`getScreenResolution()`) から返されるドット / インチの値を元に計算している。ただし、Java のメソッド (`getScreenResolution()`) の戻り値がディスプレイの物理的なサイズや解像度には追従しないため、PreSerV では、`psvJavaScreen` クラスに 1 インチあたりのドット数を外部から指定可能なメソッド (`setScreenResolution(int dpi)`) を用意しております。

(2)プリンタの場合

Java の標準メソッドでは利用しているプリンタの解像度が取得できないため、72 ドット / インチを基準とした解像度でスケールの変換を行なっています。(Java1 テクノロジー) そのため、PreSerV では印刷時に解像度を指定して印刷する機能を用意しております。

Java VM のバージョンにより指定方法が異なります。

(JDK1.1.X 利用の場合)

`ext` パッケージ `psvJavaPrinter` クラスコンストラクタに指定可能。

(Java2 利用の場合)

`psvJ2PrinterEx#set_print_scale` メソッドにて 72dpi 単位のスケール指定が可能。

`psvJ2PrinterEx#set_print_scale(2)`; とした場合は、144dpi で変換を行ないます。

注意) 現状の印刷機能では、用紙サイズや印刷の解像度により、大量なメモリを必要とします。

1.26 スケール値 (アスペクト比) について

Ver.3

Ver.4

Ver.5

Q DC WC DC の変換で座標が増えていくのはなぜでしょうか？

描画 (スケール変更)

(1)スケール初期値(8216)

---scale 8216

始点 : WC=(89.13043,227.17392),DC=(317,367)

終点 : WC=(843.4783,229.34782),DC=(665,367)

この時点では DC 値が同じなので画面上は直線になる。

(2)スケール変更(8216 3286)

---scale 3286

始点 : WC=(89.13043,227.17392),DC=(19,426)

終点 : WC=(843.4783,229.34782),DC=(887,423)

ここでは、スケールが大きくなった分

(3)スケール変更(8216 3286)

---scale 8216

始点 : WC=(89.13043,227.17392),DC=(311,316)

終点 : WC=(843.4783,229.34782),DC=(659,315)

この段階で 1 ドットずれているので描画が壊れる。

A 考えられる原因は 2 つあります。

(1) DC と WC の縦横比率が違う。

スケール値による表示領域の変更を行った際は、DC(キャンバスの物理的なサイズピクセル値)と WC(グラフィックウィンドウの表示領域として `psvJavaScreen#view()` で指定した領域)の縦横のアスペクト比を補正する処理が実行されます。

もし初期表示時に `psvJavaScreen#set_scale()` で画面表示領域を指定していない場合は、DC と WC の縦横比率が異なる場合があります。

この場合 `psvJavaScreen#set_scale()` をコールした際に X 方向のサイズでスケール値にあった表示領域を計算し、アスペクト比を一致させるため Y 方向の表示領域の補正が実行されます。

問題の現象が発生するときの WC の表示領域を出力してください。

これを確認すれば(1)が原因か判明します。

(2)スケール値はスケール分母の整数値である

もし、psvJavaScreen#view()でWCの表示領域を設定していたとすると、その後psvJavaScreen#get_scale()で取得されるスケール値は小数点以下の値が切り捨てられています。

例) 1/2000は実は1/2000.5など

その後psvJavaScreen#set_scale()によるスケール値変更を行ったとすると、この小数点以下の部分で小さな誤差が発生します。

この現象を回避するには、psvJavaScreen#view()実行後、psvJavaScreen#get_scale() + psvJavaScreen#set_scale()を実行し、取得されるスケール値と、実際のスケール値が誤差無く一致させることで回避可能です。

1.27 地図表示範囲（最小／最大）について

Ver.3

Ver.4

Ver.5

Q PreSerV における表示範囲（最小／最大）はどのように扱われているのでしょうか？

A PreSerV での表示範囲の判定は、以下の判定条件で行なっております。

【判定条件】

$\text{min} \leq \text{value} < \text{max}$

したがって、100 から 2500 までを表示範囲にしたい場合は、上下限値を以下のように設定します。

min=100

max=2501

上記判定を行なっているクラスとして以下のクラスがあります。

psvAnyMap：地図の表示スケール範囲

psvLayer：レイヤに登録された図形の表示スケール範囲

1.28 イメージシンボルについて

Ver.3

Ver.4

Ver.5

Q イメージシンボルが地図上に表示されません

A 以下の確認を行ってください。

- ・ svImageManager にイメージのキーと参照先 URL を指定する場合の参照先の URL パスは正しいでしょうか？
- ・ ブラウザ等を利用して URL 先のイメージが参照できますでしょうか？

1.29 イメージシンボルについて 2

Ver.5

Q Ver5 バイナリフォーマットから読み込んだイメージシンボルを移動後、Ver5 バイナリフォーマット形式で登録すると移動しません。

A Ver5 バイナリフォーマットから、PreSerV の図形に変換した場合、psvBinaryInfo (Ver5 バイナリ特有属性情報) に座標が設定される図形として psvImageSymbol クラスがあります。Ver5 バイナリフォーマットでは、psvBinaryInfo に登録されている情報を優先して出力するため、psvImageSymbol 図形の座標が変化する操作を行った場合は、psvBinaryInfo 内の psvBRectangle を初期化(null 設定)するか psvBinaryInfo を初期化する必要があります。

psvBinaryInfo の取得は、psvShape#get_values("BinaryInfo")でオブジェクトを取得します。設定されていない場合は、null が返ります。

psvBinaryInfo を初期化した場合は、以下のデフォルト値が設定されます。

【Ver5バイナリ特有属性情報出力】

クラス名	属性情報	バイナリフォーマット出力情報
psvBinaryInfo	形状コード	なし。 ただし、「長方形」、「環」、「円(3点指定)」、「弧(3点指定)」の場合は、それぞれの形状コードでのバイナリ出力を行う。
	レイヤ文字列	設定されている場合は、設定値のレイヤ文字列 設定されていない場合は、レイヤ番号の文字列
	シンボル名称	設定されている場合は、設定値のシンボル名称 設定されていない場合は、シンボル番号の文字列
	画像URL文字列	設定されている場合は、形状コード 114 でバイナリ出力を行う。
	文字スタイル	psvTextStyle の内容
	画像変換	psvBRectangle の内容 (*1)
	表示制限	psvBDisplayLimit の内容
psvTextStyle	事象位置	設定されている場合は、形状コード 112 でバイナリ出力を行う。
	表示幅	設定されている場合は、設定値の出力設定されていない場合は、psvTextContext#tx_width x 文字数 (半角は、半分の大きさ) を出力
	表示高さ	設定されている場合は、設定値の出力設定されていない場合は、psvTextContext#tx_height

1 技術的な質問

	フォントスタイル	設定されている場合は、設定値の出力 設定されていない場合は、1: normal を出力
	配置	設定されている場合は、設定値の出力 設定されていない場合は、2: lowerleft を出力
psvBRectangle	変換元左下座標 X Y	設定されている場合は、設定値の出力 設定されていない場合は、0 を出力
	変換元右上座標 X Y	設定されている場合は、設定値の出力 設定されていない場合は、Image オブジェクトの幅を出力
	変換先左下座標 X Y	設定されている場合は、設定値の出力 設定されていない場合は、矩形の左下 X 座標
	変換先右上座標 X Y	設定されている場合は、設定値の出力 設定されていない場合は、矩形の右上 X 座標
psvBDisplaylimit	表示サイズ	設定されている場合は、設定値の出力 を行い、バイナリフォーマット表示種別に 2: サイズ固定を出力
	表示下限	設定されている場合は、設定値の出力 を行い、バイナリフォーマット表示種別に 3: 表示下限設定を出力

1.30 DM 地図で線が表示されない現象について

Ver.5

Q MapViewer を表示しようとしているのですが、PolyLine が表示されていないようなのですが、なぜでしょうか？
何か設定が抜けているのでしょうか？

A PSV バイナリ変換ツールで、線色を設定している部分が欠如している可能性があります。
設定されていない場合、常に白色で描画されてしまいます。
DMConvert#set_attrib()メソッドで、描画色に黒色を設定することで回避できます。

リスト 15 コーディング例 (描画色に色を設定するクラス)

```
// 描画色に色を設定するクラス例

public class DMConvert implements psvConvert{

    . . . (省略) . . .

    // 指定レイヤ番号のコンテキストオブジェクトに書き換える
    public psvContext set_attrib (int layer, psvContext cx) {
        cx.set_color( new psvColor( Color.black ) );
        return cx ;
    }

    . . . (省略) . . .
}
```

1.31 DM 地図表示時のユーザー線種エラーについて

Ver.5

- Q** DM 地図を表示しているのですが、表示されると下記のようなエラーが大量に出力されてしまいます。

There is data which is not convertible. (Error Data = “ 任意の数字 ”)

This Shape is not supported yet. (code = 21)

設定が抜けているのでしょうか？

- A** エラーのコードは図形エラー：ユーザー線種(21)です。
ユーザー線種定義ファイル(AppLines.def)を登録していない可能性があります。
初期処理を確認し、該当ファイルの登録が行われていない場合は、psvMdmMapEx 生成の前に、下記のスタティックメソッドを使用し、該当ファイルを追加登録して下さい。

```
psvAppLines.load_def_file ("ユーザー線種定義ファイル");
```

1.32 Raster 地図使用時のバージョン変更で発生する、クラス未取得エラーについて

Ver.5

- Q** Raster 地図を使用しているのですが、PreSerV のバージョンを ver.4 から ver.5 に変更したところ、

```
java.lang.NoClassDefFoundError:  
jp/co/melco/preserv/def/psvDefaultRasterMap
```

上記のようなエラーが表示され実行できませんでした。

psvDefaultRasterMap クラスが存在していないのでしょうか？

- A** ver.5 では preserv50.jar から削除されています。
サンプルとして公開する予定ですので、必要であればサポートにご連絡ください。

1.33 図形読み込み失敗の原因調査方法について

Ver.3

Ver.4

Ver.5

- Q** ブラウザを再起動し同じ地域を表示しても図形が表示されません。
このような現象は、何が原因なのでしょうか？
また、原因を調査する場合どこを調べるとよいのでしょうか？

- A** 図形が表示されない原因にも様々に存在します。

プログラムの問題、サーブレットの問題、クライアントの問題を分けて、
下記の手順で検索系の処理が呼ばれているか確認してください。
データベースには図形が登録されている状態とします。

- * 注：サーブレットの問題については、MapDataManagerQ&A 集「図形
読み込み失敗の原因調査方法について」を参照してください。

1. プログラムの確認

- a) 下記のように MdmMap のマップ属性に AUTO_LOAD(自動読み込み)
がコンストラクタで指定されている事を確認してください。

```
int attrib = (psvAnyMap.MAP_VISIBLE |  
             psvAnyMap.MAP_AUTO_LOAD);
```

2. クライアント側の処理の確認

- a) addMap している screen が正しいか確認してください。
(2 つ screen がある場合、違う screen に addMap していませんか。)

- b) 下記の手順でクライアントの図形取得を確認してください。
クライアントの Java コンソールに例外が出力されていないか確認します。
psvMdmMapEx 内に登録されているメッシュを確認します。
psvMdmMapEx#array で取得できる Vector 要素に登録されている psvTree
オブジェクトで psvTree#get_mesh_no()を確認します。
以下の例のように、psvMdmMapEx 内の図形を全件検索してください。

リスト 16 全 Shape データ情報取得の方法

```
-----  
psvRegionSelect select = new psvRegionSelect (    null,  
                                                new psvRegion ( -1*Float.MAX_VALUE,  
                                                            -1*Float.MAX_VALUE,  
                                                            Float.MAX_VALUE,  
                                                            Float.MAX_VALUE));  
  
psvSearch search = new psvSearch (select, false);  
if (mdm_map.search (search)) {  
    Enumeration e = search.elements(0);  
    while (e.hasMoreElements()) {  
        psvShape shape = (psvShape)e.nextElement();  
        int mdm_key = mdm_map.get_mdm_key(shape);  
        System.out.println("mdm_key["+mdm_key+  
                            "]/ psvShape["+shape+"]");  
    }  
}  
-----
```

検索できる場合、図形の登録(検索)まではできています。
図形の設定(attrib)を確認してください。

1.34 レイヤー番号未設定の psvLayerAttrib 設定について

Ver.3

Ver.4

Ver.5

Q Layer の attrib でレイヤー番号=0 番(未設定)の表示設定ができませんでした。
何か設定方法はありますか？

A レイヤー番号が未設定の図形は Layer テーブルには登録されていないので、
psvLayerAttrib では設定できません。
下記のように、全 Shape データを取得して設定を行ってください。

リスト 17 全 Shape データレイヤー表示設定方法

```
-----  
psvRegionSelect select = new psvRegionSelect ( null,  
                                                new psvRegion ( -1*Float.MAX_VALUE,  
                                                            -1*Float.MAX_VALUE,  
                                                            Float.MAX_VALUE,  
                                                            Float.MAX_VALUE));  
psvSearch search = new psvSearch (select, false);  
if (mdm_map.search (search)) {  
    Enumeration e = search.elements(0);  
    while (e.hasMoreElements()) {  
        (psvShape)e.nextElement().attrib("設定値");  
    }  
}  
-----
```

1.35 Raster 地図表示で OutOfMemory エラーが発生してしまう現象について

Ver.5

Q Raster 地図を表示し、スクロールして画面から消えた後の解放ができず、メモリオーバーになります。
削除処理を入れているのですが、削除されていないようです。
このような場合、どうすれば宜しいでしょうか？

A メッシュ最大数の登録ができていない場合は、削除処理を行いません。
下記のように、attrib で最大数の登録を行ってください。

```
attrib( psvAnyMap.MAP_VISIBLE |  
       psvAnyMap.MAP_AUTO_LOAD |  
       meshNum << 8);
```

又、別の場所で attrib を更新していないか確認してください。
更新をした場合に、"meshNum"が 0 になってしまうと、削除されなくなります。

1.36 psvMdmMapEx から psvLayer が取得できない現象について

Ver.4

Ver.5

- Q** psvMdmMapEx にて、コンストラクタ時にレイヤーを登録しているのですが、取得してみると null になってしまいます。

```
mdm_map = new psvMdmMapEx (  
    null,  
    (psvAnyMap.MAP_VISIBLE |  
     psvAnyMap.MAP_AUTO_LOAD),  
    -1,  
    1000000,  
    new psvDefaultTransfer(),  
    (psvMapIF)op_map.elementAt(0),  
    layer,      レイヤーを設定している  
    prop,  
    psvMdmMapEx.BIG_SCALE,  
    keiNum) ;  
  
System.out.println("layer[" + mdm_map.layer() + "]);
```

出力 :

```
layer[null]
```

- A** psvMdmMapEx では、コンストラクタに指定した、レイヤーオブジェクトがデフォルトで設定されないため、以下のようにレイヤーの再設定を行ってください。

```
mdm_map.layer(layer);
```

1.37 印刷時の ScreenDraw メソッドが複数回呼び出される

Ver.3

Ver.4

Ver.5

Q jp.co.melco.preserv.j2.psvPrinterDraw#ScreenDraw()メソッドが複数回呼び出されることがある。

A Java 2DTM API プログラマーズガイドに以下の記述があります。
ScreenDraw()メソッドは、Printable#print()メソッド内でコールされます。
この Printable#print()メソッド自体が複数回コールされていることが原因と思われる。
赤太字部分が本件に関係している部分であり、Java 内部のレンダリング要求の回数分 Printable#print()メソッドが呼び出されます。

- - Java 2D™ API プログラマーズガイドより抜粋 - -

7.2.3 Printable ジョブと Pageable ジョブ Printable ジョブは、印刷を行うもっとも簡単な方法です。使うページペインタは 1 つだけで、アプリケーションは Printable インタフェースを実装する単一のクラスを提供します。印刷を行うときは、印刷システムがページペインタの print ()メソッドを呼び出して、各ページをレンダリングします。ページのインデックスは 0 から始まり、ページは順番どおりに要求されます。

ただし、次のページに進む前に、同じページのレンダリング要求が何回もページペインタに対して行われる場合があります。

最後のページが印刷されると、ページペインタの print ()メソッドは NO_SUCH_PAGE を返します。

- - Java 2D™ API プログラマーズガイドより抜粋 - -

1.38 レイヤを使った表示切替について

Ver.3

Ver.4

Ver.5

Q 同一レイヤ内に存在する線やシンボルを、特定の属性情報を元に、表示する種別を選択することはできないのでしょうか？

A psvLayer クラスの場合、図形の描画属性・可視性に不可視を設定することで個別の図形を非表示にします。

質問の動作を実現するには、特定の属性情報を持った図形の検索 描画属性の再設定 再描画が必要になります。

表示非表示の最小限の単位でレイヤを分けて、レイヤの ON/OFF で対応されることを推奨します。

1.39 Java 2 を利用した点線描画について

Ver.3

Ver.4

Ver.5

Q 点線の描画がうまくいきません。
java は、JSDK 1.4.1_03 を使用しています。

A スクリーンクラスを j 2 対応パッケージの
jp.co.melco.preserv.j2.psvJavaScreen2D クラスに
変更していただくことで点線を利用することが出来ます。

変更例

JDK コーディング例

```
psvJavaScreen screen;  
screen = new psvJavaScreen(canvas, xx, xx, xx, xx);
```

java2 利用コーディング例

```
psvJavaScreen2D screen;  
screen = new psvJavaScreen2D(canvas, xx, xx, xx, xx);
```

1.40 表示画面が反転表示される

Ver.3

Ver.4

Ver.5

- Q 地図上での図形操作後に、地図全体が反転表示されるようになった。
- A 上記のように反転表示される場合、図形操作の処理で `screen#set_raster()` メソッドの `screen#set_raster(psvScreen.GXcopy, true);` を利用しているか確認します。
- オフスクリーンに対して、`screen#set_raster(psvScreen.GXxor, true);` を呼び出したままの場合は、反転表示されます。
- 後処理として、`screen#set_raster(psvScreen.GXxor, false);` で X O R モードの解除を行います。

通常図形を移動させるような場合の操作として

以下のようなコーディングを行います。

以下のコーディング例は、Tutorial 3 章 6 (矩形のラバーバンドを描くには) でも確認できます。

リスト 18 コーディング例 (矩形のラバーバンドを描くには)

```
screen.set_paint_type (0);
screen.set_offscreen_mode(false);    //オフスクリーンモードセット
screen.set_raster (screen.GXxor,true); //XOR モードセット
screen.set_color(new psvColor(Color.green)); //色セット

if(p2 != null)
    screen.rectangle(p.x, p.y, p2.x, p2.y); //残像削除

p2 = screen.dc_to_wc (e.getX(),e.getY());
screen.rectangle(p.x, p.y, p2.x, p2.y); //図形の描画
screen.set_raster (screen.GXcopy,false); //XOR モード解除
screen.set_offscreen_mode(true);    //オフスクリーンモード解除
```


1.41 地図の回転について

Ver.3

Ver.4

Ver.5

- Q 地図回転後のスクロールは可能でしょうか？
- A 現在のところ、psvJavaScreen#rotate()メソッドで地図を回転させた状態で、スクロールを行うことはできません。
回転後は、地図上での操作を行えないようにする必要があります。

1.42 psvBinaryInfo (Ver5 バイナリ特有属性情報) について

Ver.5

Q PreSerV for Web データフォーマット Version5.0 の「注釈 (事象位置なし) 」の「表示種別」、「表示下限」、「注釈位置座標 X」、「注釈位置座標 Y」を図形オブジェクトから取得する方法を教えてください。

A PreSerV for Web データフォーマット Version5.0 は、
PreSerV で管理している図形属性情報と、G-XML の仕様に従った情報があります。

ご質問にありました下記の属性の分類は以下になります。

「注釈位置座標 X」、「注釈位置座標 Y」 =

「表示種別」「表示下限」 =

については図形の標準メソッドで取得し、については、

「psvBinaryInfo (Ver5 バイナリ特有属性情報) 」として、

各図形オブジェクトの psvArgs オブジェクトにキー名"BinaryInfo"として
オブジェクト登録しております。

psvBinaryInfo に含まれる情報は、データフォーマットの読み込み時に設定され、データフォーマットの出力時に設定されている情報を利用して出力を行ないます。

これらの情報は、図形の描画には反映されません。

リスト 19 コーディング例 (psvBTextStyle, psvBDisplayLimit 取得)

```
psvBinaryInfo bi = psvShape#get_values("BinaryInfo");  
// 文字スタイルの取得  
psvBTextStyle bt = bi.getTextStyle();  
// 表示制限の取得  
psvBDisplayLimit bd = bi.getDisplayLimit();
```

以下に「注釈」の各属性について、管理しているオブジェクトの詳細を示します。

また、注釈以外の図形で「psvBinaryInfo (Ver5 バイナリ特有属性情報) 」で管理している属性も合わせて示しています。

注釈の属性について

内容	分類	管理オブジェクト	管理オブジェクト
形状コード	PSV	psvBinaryInfo	-
総バイト数	-	-	-
レイヤ文字バイト長	-	-	-
レイヤ文字列	G-XML	psvBinaryInfo	-
表示幅	G-XML	psvBinaryInfo	psvBTextStyle
表示高さ	G-XML	psvBinaryInfo	psvBTextStyle
表示色	PSV/G-XML	psvDrawContext	psvColor
ベースライン角度	PSV/G-XML	psvDrawContext	psvTextContext
フォント種別	PSV/G-XML	psvDrawContext	psvTextContext
フォントスタイル	G-XML	psvBinaryInfo	psvBTextStyle
配置	G-XML	psvBinaryInfo	psvBTextStyle
表示種別	-	-	-
表示サイズ指定	G-XML	psvBinaryInfo	psvBDisplayLimit
表示下限	G-XML	psvBinaryInfo	psvBDisplayLimit
注釈位置座標 X	PSV/G-XML	psvShape	psvPoint
注釈位置座標 Y	PSV/G-XML	psvShape	psvPoint
注釈内容バイト数	-	-	-
注釈内容文字列	PSV/G-XML	psvDrawContext	psvTextContext

事象位置ありの場合

事象位置座標 X	G-XML	psvBinaryInfo	psvBTextStyle
事象位置座標 Y	G-XML	psvBinaryInfo	psvBTextStyle

その他 psvBinaryInfo で管理する属性

図形	属性	管理オブジェクト	管理オブジェクト
シンボル	シンボル名称	psvBinaryInfo	-
画像(URL 指定)	画像 URL 文字列	psvBinaryInfo	-
画像(URL/キ- 共通)	画像変換(元/先)	psvBinaryInfo	psvBRectangle

1.43 透過率を指定した直接描画について

Ver.3

Ver.4

Ver.5

- Q** psvJavaScreen#dc_polygon()メソッドを利用し、描画したポリゴンの塗潰しを透過描画しても透過されない現象が発生しております。

具体的には、スクリーン（psvJavaScreen2D）に対して

```
screen.set_color(new psvColor(Color.white))
screen.set_paint_type(1 | (128 << 8) | (1 << 16))
screen.dc_polygon(xs, ys)
```

といった感じでポリゴンを塗り潰そうとしているのですが、透過されずにベタ塗りとなってしまいます。

透過表示させるために、set_paint_type()以外に必要な処理がありましたらご教示ください。

ちなみに、contextの塗り潰し属性に透過率を設定したpsvShapeでは正しく透過表示されています。

- A** 下記のように直接スクリーンのメソッドを使って図形を描画する場合は、表示色の設定に直接透過色を指定してください。
以下のように設定します。

```
screen.set_color(new psvColor(new Color(255,255,255,128)))
screen.set_paint_type(1)
screen.dc_polygon(xs, ys)
```

塗潰し属性は、図形オブジェクトに設定した場合に有効となります。

1.44 レイヤの縮尺による表示切替について

Ver.3

Ver.4

Ver.5

Q 縮尺に応じて、レイヤの表示/非表示を切り替えたいと考えております。
チュートリアルを見ますと、マップ単位での表示/非表示の説明がなされていますが、レイヤ単位で縮尺に応じた表示/非表示を切り替えることはできるでしょうか？

例として

- ・線路レイヤ
- ・道路レイヤ
- ・家レイヤ

という3つのレイヤがあるとします。

線路レイヤの図形は縮尺が 1/500 ~ 1/10000 の時だけ表示する。

道路レイヤの図形は縮尺が 1/100 ~ 1/5000 の時だけ表示する。

家レイヤの図形は縮尺が 1/1 ~ 1/100 の時だけ表示するといったことを行いたいのです。

A 表示縮尺に応じてレイヤの表示 / 非表示を設定するには、`psvLayer#set_range()`メソッドを利用します。

===`psvLayer#set_range()`メソッド(JavaDocより引用)=====

```
public void set_range(int no,  
                      int amin,  
                      int amax)
```

レイヤ番号ごとの表示スケールの表示上下限值を設定します。

パラメータ:

no - レイヤ番号

amin - 表示スケール最小値

amax - 表示スケール最大値

=====

下記のような場合には以下のように設定いたします。

レイヤ情報を以下とした場合。

[レイヤ内容 (表示する縮尺) = レイヤ番号]

1 技術的な質問

- ・ 線路レイヤ (1/500 ~ 1/10000) =1
- ・ 道路レイヤ (1/100 ~ 1/5000) =2
- ・ 家レイヤ (1/1 ~ 1/100) =3

```
psvLayer layer = new psvLayer();  
// 線路レイヤ  
layer.set_range(1, 500, 10001)  
// 道路レイヤ  
layer.set_range(2, 100, 5001)  
// 家レイヤ  
layer.set_range(3, 1, 101)
```

上記以外のレイヤ情報の設定には、psvLayer クラスの JavaDoc も参照願います。

1.45 レイヤの設定可能属性について

Ver.3

Ver.4

Ver.5

Q レイヤ単位に設定できる図形の属性項目について、Javadoc を参照すると図形の

- ・色
- ・塗りつぶし属性

はレイヤ単位で設定が可能であることが分かるのですが、

- ・線種
- ・サイズ
- ・フォント(図形が `psvFontData` の場合)

などを、レイヤ単位に設定することはできないのでしょうか。

A

- ・線種
- ・フォント(図形が `psvFontData` の場合)

については、レイヤ単位の設定は行えません。

- ・サイズ

については、「シンボル、イメージシンボル、文字」図形に限り、`psvLayer#set_obj_range()`メソッドを利用することによりピクセル単位での表示上下限を設定することが出来ます。(文字については、1文字あたりの大きさの指定になります。)

このメソッドは、

- ・上下限値に同じ値を指定して、レイヤ単位で図形のサイズを固定にする。
(縮尺に関係なく固定のサイズで表示する場合)
- ・上限値に-1を指定して、レイヤの下限値を設定する。
(指定した値以上縮小されないようにする場合)
- ・下限値に-1を指定して、レイヤの上限値を設定する。
(指定した値以上拡大されないようにする場合)

のように使用することができます。

レイヤ単位で設定可能な属性として以下のものがあります。

- ・可視 / 不可視
- ・表示色
- ・塗りつぶし属性

1.46 同一座標の図形が複数ある場合の優先順位について

Ver.3

Ver.4

Ver.5

- Q** マップ内に同じ点情報をもつポリゴン、ポリライン等の図形をサーチした場合に、返す図形の優先順位はあるのでしょうか？
- A** 最初に見つかった図形になります。また、最初に見つける図形については、規則性がないためどちらの図形になるかはわかりません。

1.47 メッシュの検索について

Ver.3

Ver.4

Ver.5

Q メッシュをまたがった図形に対してのサーチを行うと図形が選択できないことがあります。全メッシュ検査する方法等ありましたら御教授願います。

A メッシュ境界を越えた座標のメッシュ操作は、仕様上制限されております。メッシュ境界については、psvTree クラスのコンストラクタもしくは、psvTree#set()メソッドにて設定 / 再設定することが出来ます。マップクラスに登録されているメッシュ情報内を、無条件で検索するコーディング例を以下に記述します。

コーディング例では、全てのメッシュ & 図形に対してチェックを行うため、性能が著しく低下することが予想されます。チェック対象とするメッシュを制限する等である程度の遅さを回避することが可能となります。

リスト20 コーディング例 (図形検索)

```
/**
 * 指定座標に一番近い図形オブジェクト情報を取得します。
 * <p>
 * @param x,y 指定座標 (WC座標)
 * @param visibleflg 可視フラグ判定有無
 * <p>
 * @return 指定座標に一番近い図形オブジェクト
 */
public synchronized psvShape flexible_search_nearest(psvAnyMap map,
                                                    float x, float y, boolean visibleflg)
{
    psvShape lastdata = null ;
    float lastdist = 100000f ;

    Enumeration vtree = map.array.elements() ;
    while(vtree.hasMoreElements()){
        psvTree tree = (psvTree)vtree.nextElement() ;
        psvRegionBounds org_rb = tree.getBounds();
        //一時的にメッシュ境界を最大にする。
        tree.set((float)Integer.MIN_VALUE, (float)Integer.MIN_VALUE,
                (float)Integer.MAX_VALUE, (float)Integer.MAX_VALUE);
        psvShape tmpdata = tree.search_nearest(x, y, visibleflg) ;
        //メッシュ境界を元に戻す。
        tree.set(org_rb.sx, org_rb.sy, org_rb.ex, org_rb.ey);
        if(tmpdata.distance(x,y)< lastdist && tmpdata != null){
            lastdata = tmpdata ;
            lastdist = tmpdata.distance(x,y);
        }
    }
    return lastdata ;
}
```

1.48 psvVectorMap#search_nearest()メソッドについて

Ver.3

Ver.4

Ver.5

- Q** 引数の psvLayer 、 int[] にはどのように値を渡せばよいのでしょうか？現状、psvLayer 、 int[]共に null をセットしていますがデフォルトとして検索されているのでしょうか？また、図形がヒットせず null が帰ってくる場合はあるのでしょうか？
- A** psvLayer オブジェクトが null の場合は、レイヤ情報を利用しない検索処理を行います。null が返される条件は、以下のものがあります。
- 1 . 指定した座標から、最も近い図形までの距離が 1 0 0 Km以上離れている場合。
 - 2 . メッシュ境界を越えた座標を指定した場合。以下の例で説明します。
- ・【メッシュ】
メッシュ A は、(0,0)-(100,100)のメッシュ境界で作成
メッシュ B は、(100,0)-(200,100)のメッシュ境界で作成
図形 1 は、(80,30)-(150,40)の矩形をもち、中心座標が(155,50)の図形
 - ・【図形登録】
図形登録時に、メッシュ境界を越えた座標を持つ図形を登録することはできません。
例：メッシュ A に対して、図形 1 の登録を行うと、エラーとなります。
中心座標が(155,50)のため、メッシュ B のみ登録されます。
 - ・【図形検索】
最近点検索 (search_nearest) では、指定された座標が、メッシュ境界内となるメッシュのみ検索対象となる。
例：座標(X,Y)=(90,30)を指定した場合、対象となるメッシュは、メッシュ A となり該当する図形は存在しない。メッシュ B については、メッシュ境界外のため、検索処理を実施しない。

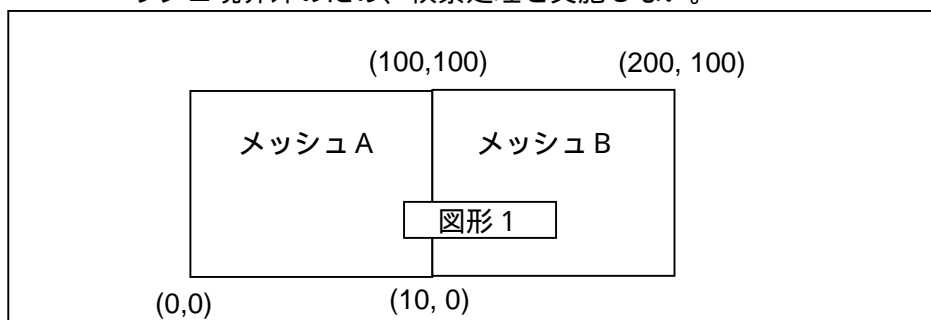


図 1 - 1 メッシュ境界

1 技術的な質問

- 3 . 独自図形等で distance() メソッドの戻り値が常に 1 0 0 K m 以上返す場合。
1 0 0 K m は PreSerV では、W C 座標上 1 0 0 0 0 0 となっております。(m 換算)

1.49 レイヤ登録数について

Ver.3

Ver.4

Ver.5

Q 扱えるレイヤ数には制限があるのでしょうか？

A レイヤを扱うための `jp.co.melco.preserv.psvLayer` クラスは、`java.util.Hashtable` クラスを利用してレイヤ情報の管理を行っているため、制限としては、`java.util.Hashtable` で扱える範囲となります。

PreSerV では、`java.util.Hashtable` オブジェクト作成時のデフォルト値には、負荷係数に 89 をセットしています。

【負荷係数について】

時間コストとスペースコストの釣り合いを取るための数値。

1.50 マップ登録数について

Ver.3

Ver.4

Ver.5

- Q スクリーンに登録するマップ数には制限があるのでしょうか？
- A jp.co.melco.preserv.psvJavaScreen クラスは、java.util.Vector クラスを利用してマップの管理を行っているため、制限としては、java.util.Vector で扱える範囲となります。

1.51 psvJavaScreen#get_wc_width()メソッドが返す WC 値について

Ver.3

Ver.4

Ver.5

Q 拡大縮小を以下の方法で行っています。

リスト 2 1 コーディング例 (拡大縮小処理)

```

/*---- 拡大 ----*/
psvRegionBounds ww = _screen.get_view();
float delta = _screen.get_wc_width(10);
_screen.view( new psvRegionBounds(ww.sx + delta,
                                   ww.sy + delta,
                                   ww.ex - delta,
                                   ww.ey - delta));

/*---- 縮小 ----*/
psvRegionBounds ww = _screen.get_view();
float delta = _screen.get_wc_width(10);
_screen.view( new psvRegionBounds(ww.sx - delta,
                                   ww.sy - delta,
                                   ww.ex + delta,
                                   ww.ey + delta));

```

この方法で拡大縮小を交互に行う (拡大 縮小 拡大) と、拡大縮小する前の縮尺と行った後の縮尺が異なります。psvJavaScreen#get_wc_width()メソッドで取得する WC 値に誤差が生じるのは何故でしょうか。また、拡大縮小を交互に行う (拡大 縮小 拡大) 場合でも、縮尺が完全に同一になるための方法はありませんでしょうか？

A psvJavaScreen#get_wc_width()メソッドでは、指定された DC 値(ドット)を WC 値に変換した値となります。そのため、表示スケール異なれば、DC 値(ドット)を表す WC 値は異なり、スケールの分母が大きければ大きいほど、指定した DC 値から取得できる WC 値は、大きな値になります。

縮尺を同一にする例として、psvJavaScreen#set_scale()メソッドに直接縮尺を指定することで拡大縮小時の縮尺を一定にします。

リスト 2 2 コーディング例 (縮尺直接指定での拡大縮小)

```

/*---- 拡大 ----*/
//現在の縮尺を取得
int scale = _screen.get_scale();
//縮尺を指定
_screen.set_scale(scale - 20);
/*---- 縮小 ----*/
//現在の縮尺を取得
int scale = _screen.get_scale();
//縮尺を指定
_screen.set_scale(scale + 20);

```

2 その他の質問

その他の質問についての Q&A です。

2.1 PreSerV for Web 署名について

Ver.3

Ver.4

Ver.5

Q PreSerV for Web の署名の種類について、ベリサインのみでしょうか？他の署名はあるのでしょうか？

A PreSerV for Web ではベリサインのみ採用しています。

2.2 GIF ライセンスについて

Ver.3

Ver.4

Ver.5

- Q** PreSerV for Web で GIF のライセンスは取得しているのでしょうか？
- A** PreSerV for Web は米国 UNISYS 社から GIF ライセンスを取得しています。
(米 Unisys の「GIF 特許」は 2004.6.20 に日本国内でも失効しています。)

