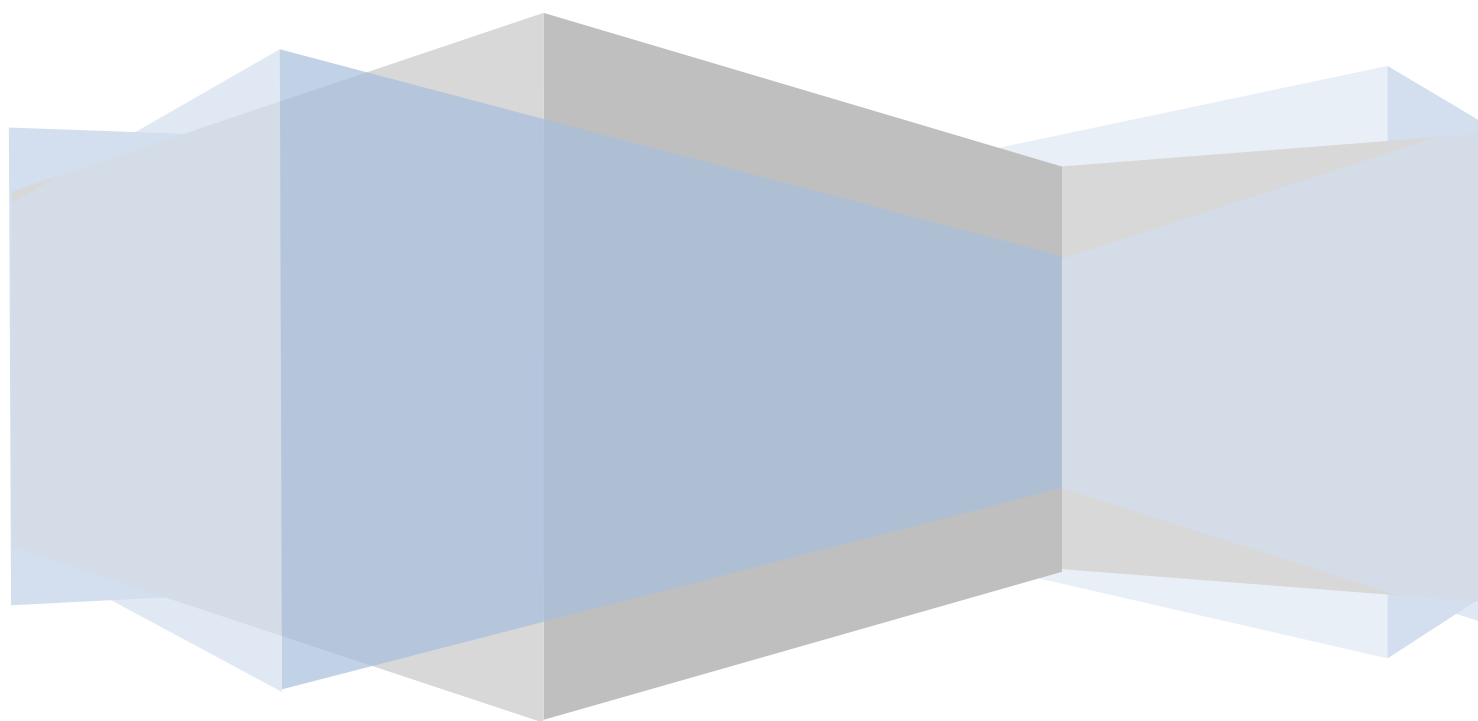


GraphWeaver for Visio

Version 1.10

Copyright 2009 (有)さくらシステム



GraphWeaver for Visio

目次

1.	はじめに.....	1
2.	ライブラリ構成.....	2
3.	APIの使用例.....	3
3.1.	DLL への参照を追加.....	3
3.2.	Visio アプリケーションインスタンスを作成.....	4
3.2.1.	可視状態起動.....	4
3.2.2.	不可視状態起動.....	4
3.3.	描画クラスを作成.....	4
3.4.	GraphWeaverMng を生成.....	4
3.5.	処理終了 Callback 関数を登録.....	4
3.6.	Visio 終了 Callback 関数を登録.....	5
3.7.	マネージ処理開始.....	5
3.8.	描画ドキュメント指定.....	6
3.8.1.	新規ドキュメント作成.....	6
3.8.2.	オープン済みドキュメント指定.....	6
3.8.3.	既存ドキュメント指定.....	6
3.9.	描画ページ指定.....	7
3.9.1.	新規ページ作成.....	7
3.9.2.	既存ページ指定.....	7
3.10.	グラフ描画処理.....	8
3.10.1.	属性データ設定.....	8
3.10.2.	グラフ描画.....	11
3.10.3.	描画結果.....	13
3.10.4.	描画グラフ保存.....	13
3.10.5.	ドキュメントのクローズ処理.....	13
4.	ページサイズ設定.....	14
5.	グラフ領域設定.....	15
6.	グラフ軸設定.....	16
6.1.	引数による設定.....	16
6.1.1.	軸の設定.....	16
6.1.2.	軸の詳細設定.....	18
6.2.	軸情報設定クラスによる一括設定.....	20
6.2.1.	軸の一括設定.....	20
7.	点属性設定.....	22
7.1.	点情報設定クラスを使用.....	22
7.2.	直接引数に設定.....	23
8.	棒属性設定.....	25
8.1.	棒情報設定クラスを使用.....	25
8.2.	直接引数に設定.....	26
9.	直接データ指定.....	27
10.	補 足.....	28

GraphWeaver for Visio

1. はじめに

GraphWeaver for Visio を使用して、グラフを描画する際の諸設定に関して、主に描画ページの設定、グラフ属性の設定等について記載しています。

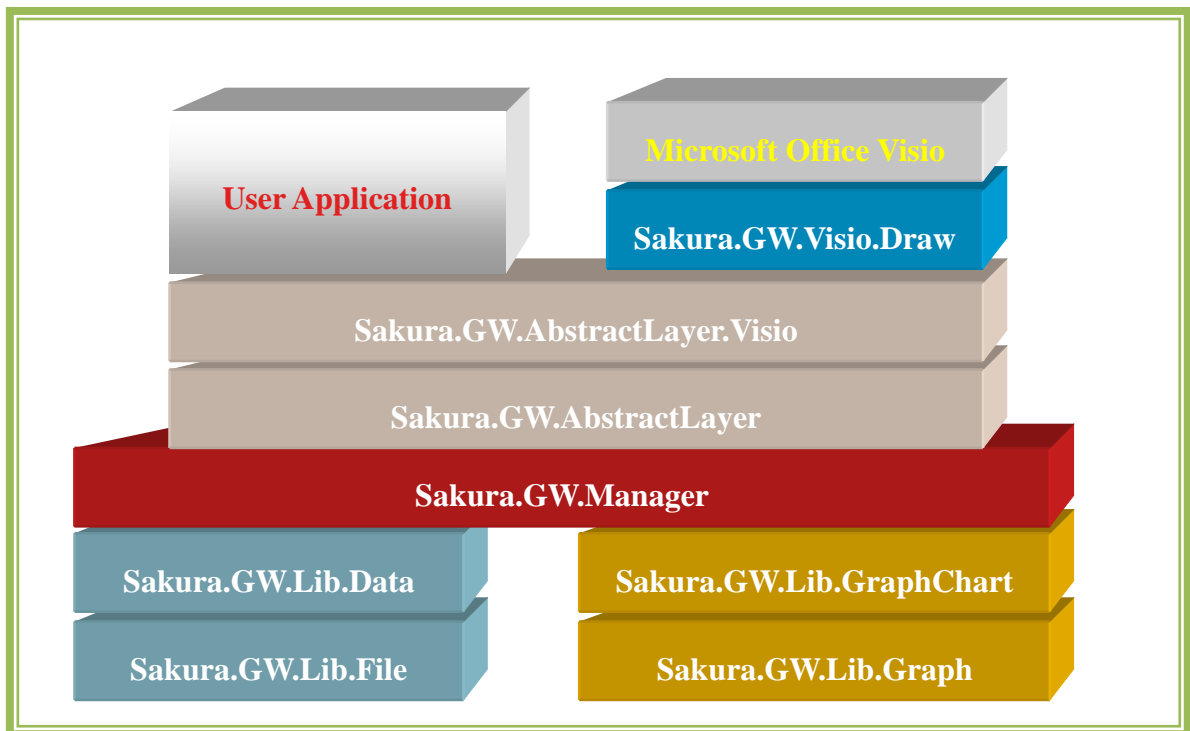
尚、本ドキュメントは GraphWeaver for Visio Ver.1.10 に関する内容です。
バージョンにより、記載内容と異なる場合がありますので、ご注意ください。

各 Namespace、クラス、メソッドの詳細については、同梱されたヘルプファイル
(GraphWeaverForVisio.chm)を参照下さい。

GraphWeaver for Visio

2. ライブラリ構成

GraphWeaver for Visio は、以下の図で示したライブラリ構成です。



- ◆ User Application
ユーザ作成アプリケーション
- ◆ Microsoft Office Visio
グラフ描画対象アプリケーション
- ◆ Sakura.GW.Visio.Draw
Visio 描画処理モジュール
- ◆ Sakura.GW.AbstractLayer.Visio
- ◆ Sakura.GW.AbstractLayer
GraphWeaver 管理モジュール
- ◆ Sakura.GW.Manager
ライブラリ管理モジュール
- ◆ Sakura.GW.Lib.GraphChart
- ◆ Sakura.GW.Lib.Graph
- ◆ Sakura.GW.Lib.Data
- ◆ Sakura.GW.Lib.File
各処理モジュール

GraphWeaver for Visio

3. API の使用例

GraphWeaver for Visio API を使用した、グラフ描画アプリケーションの構築手段です。ここでは、Visual Studio 上の C# による例を記載します。

3.1. DLL への参照を追加

プロジェクトの参照設定に参照の追加を行い、以下の DLL ファイルを追加してください。

- ◆ Sakura.GW.AbstractLayer.dll
- ◆ Sakura.GW.AbstractLayer.Visio.dll
- ◆ Sakura.GW.Lib.Data.dll
- ◆ Sakura.GW.Lib.dll
- ◆ Sakura.GW.Lib.FileLib.dll
- ◆ Sakura.GW.Lib.Graph.dll
- ◆ Sakura.GW.Lib.GraphChart.dll
- ◆ Sakura.GW.Manager.dll
- ◆ Sakura.GW.Visio.Draw.dll

GraphWeaver for Visio

3.2. Visio アプリケーションインスタンスを作成

Visio2007 SDK を使用し、Visio.Application クラスのインスタンスを生成します。このタイミングで Visio が起動します。詳細に関しては、上記の SDK に付属する資料等をご参考ください。

3.2.1. 可視状態起動

Visio が可視状態で起動します。

e.g.)

```
Visio.Application application = new Microsoft.Office.Interop.Visio.ApplicationClass();
```

3.2.2. 不可視状態起動

Visio が不可視状態で起動します。可視状態にするには、Visible プロパティを true に変更します。

e.g.)

```
Visio.Application application =
```

```
    (Visio.Application)new Microsoft.Office.Interop.Visio.InvisibleAppClass();
```

```
application.Visible = true; //可視状態に変更
```

3.3. 描画クラスを作成

3.2.で作成したアプリケーションハンドルを引数に設定し、VisioAddInLib_draw クラスのインスタンスを生成します。

```
VisioAddInLib_draw(Application application);
```

```
Application : Microsoft Office アプリケーションハンドル
```

e.g.)

```
Visio.Application visioApp = new Visio.Application();
```

```
visioDrawModule = new VisioAddInLib_draw(visioApp);
```

3.4. GraphWeaverMng を生成

3.3.で作成したインスタンスをコンストラクタ引数に設定し、GraphWeaverMng クラスのインスタンスを生成します。

```
GraphWeaverMng(IVisioDrawAccess drawClass);
```

e.g.)

```
VisioAddInLib_draw visioDrawModule; //Visio 用描画モジュール
```

```
GraphWeaverMng visioManager; //GraphWeaver 管理マネージャ
```

```
visioManager = new GraphWeaverMng(visioDrawModule);
```

3.5. 処理終了 Callback 関数を登録

3.3.で作成したインスタンスの ResponseFunc プロパティに、処理終了通知時にコールバックされる Callback 関数を登録します。

e.g.)

```
visioManager.ResponseFuncs +=
```

```
    new GraphWeaverMng.callbackResponse(CallbackFunc);
```

```
// GraphWeaver 処理終了時のコールバック登録
```

GraphWeaver for Visio

3.6. Visio 終了 Callback 関数を登録

3.3.で作成したインスタンスの `VisioClosedFunc` プロパティに、Visio 終了通知時にコールバックされる `Callback` 関数を登録します。

コールバックされた時に、上記で取得した Visio アプリケーションのインスタンスが手動で終了または外部から終了されたことが分かるので、以後インスタンスへのアクセスをしないようにしてください。

e.g.)

```
visioManager.VisioClosedFunc +=  
    new GraphWeaverMng.callbackVisioClosed(CallbackFunc);  
// Visio 終了時のコールバック登録
```

3.7. マネージ処理開始

3.3.で作成したインスタンスの `start` メソッドを呼び出すことにより、マネージ処理を開始します。

e.g.)

```
visioManager.start(); //マネージ処理開始
```

GraphWeaver for Visio

3.8. 描画ドキュメント指定

グラフを描画するページを含むドキュメントを指定します。ドキュメントを指定する方法は3つあるので、下記の中から適切な方法で処理してください。

GraphWeaver では、ドキュメントの管理を Index 値(long 型の整数)で行うので、処理中のドキュメントに対応する Index 値を次の描画ページ指定において使用します。

この操作で指定したドキュメントを閉じて開く時には、再度指定が必要です。

3.8.1. 新規ドキュメント作成

GraphWeaverMng インスタンスの createDocument メソッドを使用します。テンプレートを使用する場合は、引数 templateDocPath にパスを指定します。

```
bool createDocument(string templateDocPath, out long index);
```

e.g.)

```
long docIndex;  
visioManager.createDocument("", out docIndex);
```

3.8.2. オープン済みドキュメント指定

既に Visio で開かれているドキュメントを対象にする場合は、GraphWeaverMng インスタンスの takeoverDocument メソッドを使用します。

引数 documentHandle には、対象とするドキュメントの Document ハンドル (Microsoft.Office.Interop.Visio.Document 型)を渡します。

引数 docName、docFullPath は、指定したドキュメントの情報で上書きされます。一度も保存していないドキュメントを指定した場合、fullPath は空文字となります。

```
bool takeoverDocument(  
    object          documentHandle,  
    out string      documentName,  
    out string      fullPath,  
    out long        index );
```

3.8.3. 既存ドキュメント指定

開かれていない既存のドキュメントファイルを対象にする場合は、GraphWeaverMng インスタンスの openDocument メソッドを使用します。

引数 fullPath にファイルのフルパスを指定し、成功した場合はドキュメントがオープンされます。

```
bool openDocument(  
    string          fullPath,  
    out string      docName,  
    out long        index );
```


GraphWeaver for Visio

3.9. 描画ページ指定

描画対象ドキュメントを上記の描画ドキュメント指定によって GraphWeaver の管理下に置いた後、描画対象ページを指定します。

ここでは、ドキュメント指定した際に取得した Index 値を使用します。

ページを指定する方法は2つあるので、下記の中から適切な方法で処理してください。

3.9.1. 新規ページ作成

GraphWeaverMng インスタンスの createPage メソッドを使用します。ページに名前をつける場合は、引数 pageName にパスを指定します。同ドキュメント内に同一名のページを作成することはできないので、注意してください。

pageName に空文字を指定した場合、任意の名前が割り振られます。

```
bool createPage(long docIndex, ref string pageName);
```

e.g.)

```
string pageName = "2次関数のグラフ";
```

```
visioManager.CreatePage(docIndex, pageName);
```

3.9.2. 既存ページ指定

存在するページを対象にする場合は、GraphWeaverMng インスタンスの takeoverPage メソッドを使用します。

引数 pageName に対象とするページ名を指定して下さい。

```
bool takeoverPage(long docIndex, string pageName);
```

e.g.)

```
string pageName = "2次関数のグラフ";
```

```
visioManager.takeoverPage(docIndex, pageName);
```

GraphWeaver for Visio

3.10. グラフ描画処理

3.10.1. 属性データ設定

グラフ描画を行う設定を、GraphWeaverData クラスのインスタンスに設定します。設定は、インスタンスの各メソッド呼び出しによって行います。

e.g.)

```
GraphWeaverData mngInfo = new GraphWeaverData();
```

◆ SetDocumentFile

『3.8.描画ドキュメント指定』で指定したドキュメントの情報を設定します。テンプレートを使用しないときは、TemplateFileFullPath 引数に空文字を指定してください。また新規ドキュメントおよび未保存ドキュメント指定の時、引数 FileFullPath は空文字になります。

```
void SetDocumentFile(  
    string      DocumentName,  
    string      FileFullPath,  
    string      TemplateFileFullPath );
```

◆ SetPage

『3.9.描画ページ指定』で指定したページの描画情報を設定します。
『4.ページサイズ設定』を参照してください。

◆ SetDataSeriesNo

グラフで使用する系列数を設定します。初期値は1で、値の範囲は1～9です。設定すると、SetDataFile および SetDataRowData の情報がクリアされるので、まずこの設定から行ってください。

```
void SetDataSeriesNo (  
    int      SeriesNo ); // 系列数
```

◆ SetDataFile

グラフで使用するデータが格納されているファイルを指定します。系列 series の値の範囲は0～(SetDataSeriesNo で指定した値-1) です。同一系列番号で SetDataRowData と併用はできません。

```
void SetDataFile(  
    int      series, // 系列番号  
    string   FileFullPath, // ファイルパス  
    GraphWeaver_DataType type ); // ファイルタイプ
```

★GraphWeaver_DataType ファイルタイプ

メンバ名	内容
None	未定義(初期値、使用不可)
CsvFile	CSV ファイル
BinaryFile	バイナリファイル(未対応)

GraphWeaver for Visio

◆ SetDataRowData

グラフで使用するデータを直接指定します。

系列 series の値の範囲は0 ~ (SetDataSeriesNo で指定した値-1) です。

同一系列番号で SetDataFile と併用はできません。

List<Point>型のデータ群を引数 pointList に指定します。Point 変数の格納方法については『9.直接データ指定』を参照してください。

```
bool SetDataRowData (
    int series, // 系列番号
    List<Point> pointList); // ポイントリスト
```

◆ SetGraphSize

ページに対するグラフ領域を設定します。

『5.グラフ領域設定』を参照してください。

◆ SetGraphTitle

グラフタイトルを設定します。表示位置はページ中央最上部に固定です。

graphTitle に空欄を指定した場合は、描画しません。

```
void SetGraphTitle(
    string graphTitle, // グラフタイトル
    GraphWeaver_FontAttribute titleAttr, // フォント属性 );
```

★GraphWeaver_FontAttribute: フォント属性

プロパティ名	型名	説明
fontSize	double	フォントサイズ 単位はページ設定に依存します
fontColor	RGB	フォントの色 ※補足を参照 setIndex メソッドで ColorIndex 指定
fontSizePoint	double	フォントサイズ 単位は pt (fontSize と内部変数を共有しています)

◆ SetGraphImageLimit

描画するグラフの象限を設定します。通常は全象限を対象にします。

```
public void SetGraphImageLimit(Quadrant quadrant);
```

★Quadrant: 象限

メンバ名	内容
QuadrantAll	全象限を対象
Quadrant1	第1象限を対象
Quadrant2	第2象限を対象
Quadrant3	第3象限を対象
Quadrant4	第4象限を対象
Quadrant12	第1・2象限を対象
Quadrant23	第2・3象限を対象
Quadrant34	第3・4象限を対象
Quadrant14	第1・4象限を対象

GraphWeaver for Visio

- ◆ **SetGraphAxis**
グラフの軸属性を設定します。
『6.グラフ軸設定』を参照してください。
- ◆ **SetGraphAxisDetail**
グラフの軸詳細属性を設定します。
『6.1.2.軸の詳細設定』を参照してください。
- ◆ **SetGraphPoint**
グラフの点属性を設定します。グラフの種類が折れ線グラフの時に有効です。
『7.点属性設定』を参照してください。
- ◆ **SetGraphBar**
グラフの棒属性を設定します。グラフの種類が棒グラフの時に有効です。
『8.棒属性設定』を参照してください。

GraphWeaver for Visio

3.10.2. グラフ描画

3.10.1.で作成した `GraphWeaverData` インスタンスの `SetCommand` メソッドに「グラフ描画」を指定し、3.4.で作成した `GraphWeaverMng` インスタンスの `DoCommand` メソッドにより、グラフ描画処理を開始します。

通常は、`drawGraphAll` コマンドを使用して描画し、中断は `Abort` コマンドを使用します。

`Void SetCommand(GraphWeaverData.CommandID ID); //コマンドのセット`

CommandID: コマンド

メンバ名	内容
None	未定義 (使用不可)
Abort	処理中断
loadData	元データ読み込み •SetDataFile で設定されたファイルからデータを読み込みます
saveData	元データへ書き込み (未対応)
eraseData	データ削除 •SetDataFile で設定されたファイルを削除します
setupGraph	グラフ書式設定 •読み込んだデータとグラフ設定を元にグラフ座標計算を行います loadData コマンド実行済みの必要があります (SetDataFile で設定した系列データの場合) •設定したデータとグラフ設定を元にグラフ座標計算を行います (SetDataRawData で設定した系列データの場合)
storeGraph	グラフ書式への保存 (未対応)
resetupGraph	グラフ書式再設定 •描画済みグラフに対し、現在保持しているデータを元に再計算を行います
drawGraphAll	グラフ一括描画 (loadData+setupGraph+drawGraph) グラフを描画するために必要なコマンドを一括で実行します
drawGraph	グラフ描画 グラフ座標計算済みのグラフを描画します setupGraph 済みの必要があります
redrawGraph	グラフ再描画 グラフ描画済みのページに対して、一度グラフを消去し再描画します deleteGraph + drawGraph と同じです
deleteGraph	グラフ削除 描画されたグラフを消去します (軸、点等がページから消去されます)
reflectGraph	表示グラフからの反映 (未対応)
drawPoint	データ点のみ描画 ページに対して、データ点およびそれに付随する要素の描画を行います
deletePoint	データ点のみ削除 描画済みページからデータ点およびそれに付随する要素の削除を行います
redrawGraphPoint	データ点のみ再計算描画 描画済みページからデータ点およびそれに付随する要素の削除を行い、再描画します deletePoint+drawPoint と同じです

GraphWeaver for Visio

```
bool DoCommand(long docIndex, GraphWeaverData mngInfo); // コマンドの実行
```

e.g.)

```
mngInfo.SetCommand(GraphWeaverData.CommandID.drawGraphAll);
```

```
visioManager.DoCommand(docIndex, mngInfo);
```

GraphWeaver for Visio

3.10.3. 描画結果

『3.5. 処理終了 Callback 関数を登録』で設定した Callback 関数が呼び出され、引数で渡される共通エラーIDにより描画結果がわかります。

ErrorID: 共通エラーID ※補足を参照

3.10.4. 描画グラフ保存

描画されたグラフをファイルに保存するには、GraphWeaverMng インスタンスの saveDocument メソッドを使用します。

引数 index には3.8.で取得した Index 値を、fullPathName にはファイルのフルパスを指定して下さい。

```
bool saveDocument (Int64 index, string fullPathName);
```

e.g.)

```
string documentName = @"c:\temp\test.vsd";
```

```
visioManager.saveDocument(docIndex, documentName);
```

3.10.5. ドキュメントのクローズ処理

開いているドキュメントを閉じるには、GraphWeaverMng インスタンスの closeDocument メソッドを使用します。

引数 index には3.8.で取得した Index 値を指定して下さい。

```
bool closeDocument (Int64 index);
```

e.g.)

```
visioManager.closeDocument(docIndex);
```

GraphWeaver for Visio

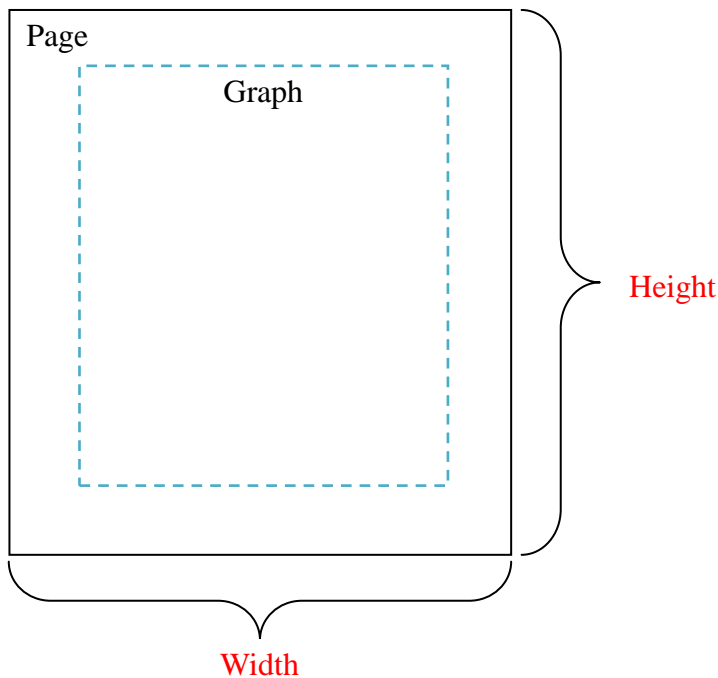
4. ページサイズ設定

GraphWeaverData::SetPage を使用して、ページサイズの設定を行います。
ここで設定した単位は、ページ内の各値の単位として使用されるので注意してください。

```
void SetPage(  
    string      pageName,  
    double      Width,  
    double      Height,  
    double      Scaling,  
    LengthUnit Unit );
```

引数名	型名	説明
pageName	string	ページ名
Width	double	ページ幅
Height	double	ページ高さ
Scaling	double	スケール係数 (通常は等倍の1.0)
Unit	LengthUnit	単位
	mm	ミリ
	inch	インチ

ex) mngInfo.SetPage(pageName, 210, 297, 1.0, LengthUnit.mm);



SetPage で指定した **Width**、**Height** がそれぞれページの幅、高さになります。
Width、**Height** の単位は、Unit の設定値に従います。

Scaling スケール係数は通常1.0にして下さい。また、ここで設定した属性に基づきページ設定を上書きするので、既存ページを指定する時は注意してください。

GraphWeaver for Visio

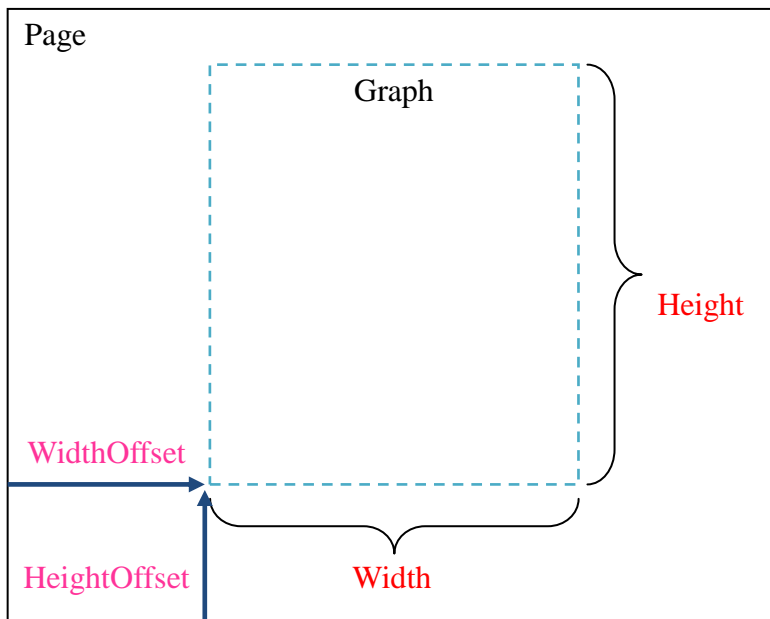
5. グラフ領域設定

GraphWeaverData::SetGraphSize を使用して、グラフ領域の設定を行います。

```
void SetGraphSize(  
    graphType    graphType,  
    double       Width,  
    double       Height,  
    double       WidthOffset,  
    double       HeightOffset    );
```

引数名	型名	説明
graphType	graphType	グラフの種類
	LineChart	折れ線グラフ
	BarChart	棒グラフ
	PieChart	円グラフ (未対応)
	CircularAreaChart	チャート (未対応)
MeterChart	動的メータグラフ (未対応)	
Width	double	グラフ幅
Height	double	グラフ高さ
WidthOffset	double	幅余白
HeightOffset	double	高さ余白

e.g.) mngInfo.SetGraphSize(GraphType.LineChart, 168.0, 237.0, 21.0, 30.0);



SetGraphSize に指定した **Width**、**Height** がグラフ領域の幅、高さに、**WidthOffset**、**HeightOffset** が、ページ原点からのオフセット量(幅・高さ)になります。本バージョンでは、ページ原点は左下になります。値の単位は、GraphWeaverData::SetPage で指定した Unit の設定値に従います。

GraphWeaver for Visio

6. グラフ軸設定

折れ線グラフ等で使用される軸の設定方法について説明します。

軸の設定方法には以下の2種類があり、どちらか一方の方法で設定してください。

- ・ 引数による設定
- ・ 軸情報設定クラスによる一括設定

6.1. 引数による設定

軸設定に関して2つのメソッドを使用します。最低限必要な軸設定項目と、色などの詳細設定項目に分けて設定が可能です。詳細設定については、省略可能です。

6.1.1. 軸の設定

GraphWeaverData::SetGraphAxis メソッドを使用して各軸の設定を行います。

```
void SetGraphAxis(  
    AxisSpec    axisSpec,  
    bool        bEnable,  
    double      min,  
    double      max,  
    double      tics,  
    int         mTics,  
    RGB         axisColor    );
```

e.g.)

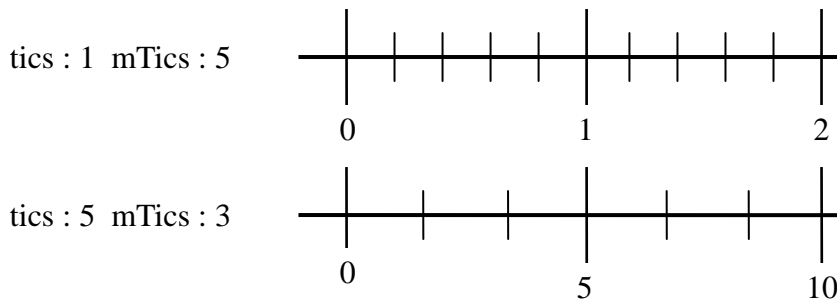
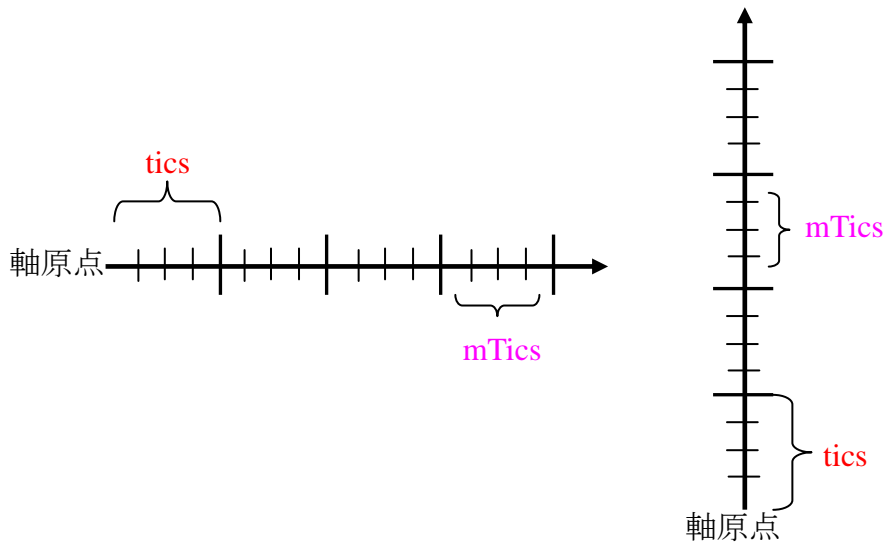
```
SetGraphAxis( AxisSpec.AxisX , true , 0 , 10 , 5 , 5 , new RGB(ColorIndex.Black) );
```

引数名	型名	説明
axisSpec	AxisSpec.AxisX	X 軸の設定
	AxisSpec.AxisY	Y 軸の設定
bEnable	bool	軸の表示/非表示の設定
min	double	軸目盛りの最小値 (省略時 : 0)
max	double	軸目盛りの最大値
tics	double	軸目盛りの間隔
mTics	int	軸目盛り間の分割数
axisColor	RGB	軸の色 ※補足を参照

GraphWeaver for Visio

★X 軸 (axisSpec = AxisX)

★Y 軸 (axisSpec = AxisY)



軸原点から、tics で指定した間隔で第1補助線が引かれます。
第1補助線間を mTics の指定値で分割し、第2補助線が引かれます。
max、min の指定値のうち絶対値が大きいほうが第1補助線の最大値となります。
bEnable に true を指定している場合のみ、X 軸が表示されます。
tics に0以下の値を指定した場合、第1補助線、第2補助線は表示されません。
mTics に0以下の値を指定した場合、第2補助線は表示されません。

GraphWeaver for Visio

6.1.2. 軸の詳細設定

GraphWeaverData::SetGraphAxisDetail メソッドを使用して各軸の詳細設定を行います。
なお、このメソッドで設定される内容は SetGraphAxis メソッドを呼び出すと初期値に設定されます。

原点「0」のサイズ、色は X 軸の TicFontSize / TicColorIndex 設定に依存しています。

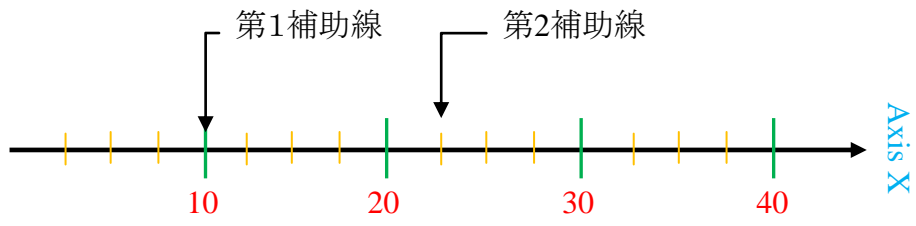
```
void SetGraphAxisDetail(  
    axisSpec      axisSpec,  
    string        title,  
    double        fontSize,  
    RGB           fontColor,  
    double        ticLength,  
    double        ticThickness,  
    RGB           ticColor,  
    double        ticFontSize,  
    RGB           ticFontColor,  
    double        mTicLength,  
    double        mTicThickness  
    RGB           mTicColor );
```

```
e.g.)  mngInfo.SetGraphAxisDetail( AxisSpec.AxisX,  
                                   "X", 8.0, new RGB(ColorIndex.Blue),  
                                   5, 0.4, new RGB(ColorIndex.Green),  
                                   5.0, new RGB(ColorIndex.Red),  
                                   2.5, 0.2, new RGB(ColorIndex.Yellow));
```

引数名	型名	説明
axisSpec	AxisSpec .AxisX	X 軸の設定
	AxisSpec .AxisY	Y 軸の設定
title	string	グラフのタイトル
fontSize	double	タイトルのフォントサイズ (単位はページサイズ設定に準ずる)
fontColor	RGB	タイトルの色 ※補足を参照
ticLength	double	第1補助線の長さ
ticThickness	double	第1補助線の太さ
ticColor	RGB	第1補助線の色 ※補足を参照
ticFontSize	double	第1補助線単位のフォントサイズ
ticFontColor	RGB	第1補助線単位の色 ※補足を参照
mTicLength	double	第2補助線の長さ
mTicThickness	double	第2補助線の太さ
mTicColor	RGB	第2補助線の色

GraphWeaver for Visio

★X 軸 (axisSpec = AxisX)



GraphWeaver for Visio

6.2. 軸情報設定クラスによる一括設定

軸設定項目を軸情報設定クラスに対して設定し、それを `GraphWeaverData::SetGraphAxisPack` により設定します。必要な設定項目を一括で設定できます。各項目の内容は6.1.で解説したものと同じです。

6.2.1. 軸の一括設定

軸の設定データを格納するための `GraphWeaver_AxisPack` クラスインスタンスを生成します。そのインスタンスに各軸属性を設定し、『3.10.1.属性データ設定』で生成したインスタンスに一括設定します。

```
e.g.) GraphWeaver_AxisPack axisPack = new GraphWeaver_AxisPack();
      axisPack.axisSpec = AxisSpec.AxisX;
      ...
      mngInfo.SetGraphAxisPack ( axisPack );
```

★GraphWeaver_AxisPack クラス

プロパティ名	型	説明
axisSpec	AxisSpec	軸種別
	AxisX	X 軸
	AxisY	Y 軸
	AxisZ	Z 軸
	Circle	円軸
bEnable	bool	有効フラグ true:軸表示, false:軸非表示
min	double	最小値
max	double	最大値
tickness	double	軸太さ
aixsColor	RGB	軸の色 ※補足を参照 setIndex メソッドで ColorIndex 指定
title	string	軸名称
titleAttr	GraphWeaver_FontAttribute	軸名称フォント属性
additionalLine1	Additional1	第1補助線属性
additionalLine2	Additional2	第2補助線属性

★GraphWeaver_FontAttribute: フォント属性

プロパティ名	型名	説明
fontSize	double	フォントサイズ 単位はページ設定に依存します
fontColor	RGB	フォントの色 ※補足を参照 setIndex メソッドで ColorIndex 指定
fontSizePoint	double	フォントサイズ 単位は pt (fontSize と内部変数を共用しています)

GraphWeaver for Visio

★Additional1クラス

プロパティ名	型	説明
tics	double	第1補助線きざみ値
ticLength	double	第1補助線長さ
ticThickness	double	第1補助線太さ
ticColor	RGB	第1補助線の色 ※補足を参照 setIndex メソッドで ColorIndex 指定
ticFontAttr	GraphWeaver_FontAttribute	第1補助線数値属性(きざみ値の数値表示属性)

★Additional2クラス

プロパティ名	型	説明
mTics	double	第2補助線分割数
mTicLength	double	第2補助線長さ
mTicThickness	double	第2補助線太さ
mTicColor	RGB	第2補助線の色 ※補足を参照 setIndex メソッドで ColorIndex 指定

GraphWeaver for Visio

7. 点属性設定

GraphWeaverData::SetGraphPoint メソッドを使用して、点属性の設定を行います。

7.1. 点情報設定クラスを使用

点情報設定クラス(GraphWeaver_PointPack)のインスタンスを作成し、点に関する情報を入力します。

```
e.g.) GraphWeaver_PointPack pointPack = new GraphWeaver_PointPack();
      pointPack.style = PointStyle.Circle;
      ...
```

★GraphWeaver_PointPack クラス

プロパティ名	型	説明
style	PointStyle	点のスタイル
	None	未定義 (使用不可)
	Circle	円
	Triangle	三角形
	Square	四角形
	Pentagon	五角形
	Hexagon	六角形
	Heptagon	七角形
	Octagon	八角形
	Star	星
Cross	十字	
pointColor	RGB	点の色 ※補足を参照 setIndex メソッドで ColorIndex 指定
size	double	点のサイズ
bConnect	bool	接続線の有無
lineColor	RGB	接続線の色 ※補足を参照 setIndex メソッドで ColorIndex 指定
lineThickness	double	接続線の太さ

設定した点情報クラスを引数として SetGraphPoint メソッドを呼ぶことにより、点属性の設定が行われます。

```
void SetGraphPoint(
    int seriesNo, //系列番号
    GraphWeaver_PointPack pack ); //点情報設定クラス
```

```
e.g.) mngInfo.SetGraphPoint( 0, pointPack );
```


GraphWeaver for Visio

7.2. 直接引数に設定

点情報設定クラスを使用せず、直接引数によって各要素の設定を行います。

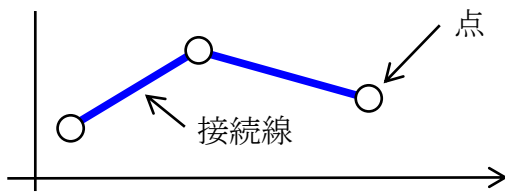
```
void SetGraphPoint(  
    int          seriesNo,  
    PointStyle   style,  
    RGB          pointColor,  
    double       size,  
    bool         bConnect,  
    RGB          lineColor,  
    double       lineThickness);
```

引数名	型	説明
seriesNo	int	系列番号
style	PointStyle	点のスタイル
	None	未定義 (使用不可)
	Circle	円
	Triangle	三角形
	Square	四角形
	Pentagon	五角形
	Hexagon	六角形
	Heptagon	七角形
	Octagon	八角形
	Star	星
Cross	十字	
pointColor	RGB	点の色 ※補足を参照
size	double	点のサイズ
bConnect	bool	接続線の有無
lineColor	RGB	接続線の色 ※補足を参照
lineThickness	double	接続線の太さ

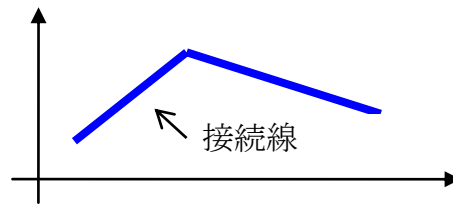
```
e.g.) mngInfo.SetGraphPoint(PointStyle.Circle, new RGB(ColorIndex.Transparence),  
                             2.0, true, new RGB(ColorIndex.Red), 0.5);
```

GraphWeaver for Visio

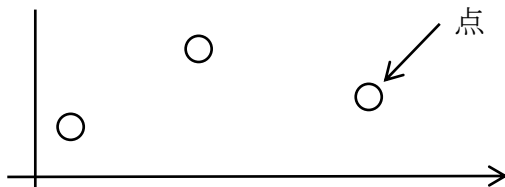
<線接続あり>



<線接続あり、点色は透明色>



<線接続なし>



style により、点の形状を設定します。指定した各形状(円、多角形、星、十字等)の中心点が、点の位置座標になります。

bConnect に **true** を指定した場合、点と点が接続線で結ばれます。

点を表示せず線だけ表示したい場合は、点の色を透明色(`ColorIndex.Transparency`)に設定してください。

GraphWeaver for Visio

8. 棒属性設定

GraphWeaverData::SetGraphBar メソッドを使用して、棒属性の設定を行います。

8.1. 棒情報設定クラスを使用

棒情報設定クラス(GraphWeaver_BarPack)のインスタンスを作成し、棒に関する情報を入力します。

```
e.g.) GraphWeaver_BarPack barPack = new GraphWeaver_BarPack();
      barPack.style= BarStyle.Normal;
      ...
```

★GraphWeaver_BarPack クラス

プロパティ名	型	説明
style	BarStyle	棒のスタイル
	None	未定義(使用不可)
	Normal	通常
	Cumulative	累積(未対応)
	CumulativeRatio	累積割合(未対応)
type	BarType	棒のタイプ
	None	未定義(使用不可)
	Vertical	縦棒
	Horizontal	横棒
barColor	RGB	棒の色 ※補足を参照 setIndex メソッドで ColorIndex 指定
width	double	棒の幅

設定した点情報クラスを引数として SetGraphBar メソッドを呼ぶことにより、点属性の設定が行われます。

```
void SetGraphBar(
    int seriesNo, //系列番号
    GraphWeaver_BarPack pack ); //棒情報設定クラス
```

```
e.g.) mngInfo.SetGraphBar( 0, barPack );
```

GraphWeaver for Visio

8.2. 直接引数に設定

棒情報設定クラスを使用せず、直接引数によって各要素の設定を行います。

```
void SetGraphBar(  
    int          seriesNo,  
    BarStyle     style,  
    BarType      type  
    RGB          barColor,  
    double       width );
```

引数名	型	説明
style	BarStyle	棒のスタイル
	None	未定義 (使用不可)
	Normal	通常
	Cumulative	累積(未対応)
	CumulativeRatio	累積割合(未対応)
type	BarType	棒のタイプ
	None	未定義(使用不可)
	Vertical	縦棒
	Horizontal	横棒
barColor	RGB	棒の色 ※補足を参照 setIndex メソッドで ColorIndex 指定
width	double	棒の幅

e.g.) `mngInfo.SetGraphBar(PointStyle.Circle, new RGB(ColorIndex.Transparence),
2.0, true, new RGB(ColorIndex.Red), 0.5);`

9. 直接データ指定

GraphWeaverData クラスのインスタンスに対し、直接系列データを指定します。使用するデータの次元によって **Point** 型コンストラクタの引数が異なるので、データに合わせて処理してください。

9.1. 1次元データ

棒グラフで使用するデータは1次元データとし、各要素順に **Point** 型変数を作成して **List** を作成します。

```
e.g.) List<Point> pointList = new List<Point>();
      foreach ( double data in Data)
      {
          Point pos = new Point(data); //1次元データとして Point 作成
          pointList.Add( pos );
      }
      mngInfo.SetDataRowData ( 0, pointList );
```

9.2. 2次元データ

折れ線グラフで使用するデータは2次元データとし、各要素 **Point** 型変数を作成して **List** を作成します。

```
e.g.) List<Point> pointList = new List<Point>();
      for ( int i = 0; i < Data.Count(); i++)
      {
          Point pos = new Point(Data[i].x, Data[i].y); //2次元データとして Point 作成
          pointList.Add( pos );
      }
      mngInfo.SetDataRowData ( 0, pointList );
```

GraphWeaver for Visio

10. 補 足

ErrorID: 共通エラーID

メンバ名	内 容
NoError	エラーなし 正常終了
UnknownCommand	不明コマンド コマンド ID が不明です
NotSupportCommand	未対応コマンド 現在サポートされていないコマンドを実行しました
UnknownDataType	データタイプ不明 不明なデータタイプが指定されました
NotSupportDataType	未対応データタイプ 未対応のデータタイプが指定されました
UnknownFileType	ファイルタイプ不明 不明なファイルタイプが指定されました
NotSupportFileType	未対応ファイルタイプ 未対応のファイルタイプが指定されました
OpenFileError	ファイルオープン失敗 指定されたファイルのオープンに失敗しました
ReadFileError	ファイルリード失敗 指定されたファイルの読み込みに失敗しました
WriteFileError	ファイルライト失敗 指定されたファイルへの書き込みに失敗しました
CreateFileError	ファイル作成失敗 指定されたファイルの作成に失敗しました
DeleteFileError	ファイル削除失敗 指定されたファイルの削除に失敗しました
DataFormatError	データフォーマットエラー 指定されたファイルの形式がデータタイプと一致しません
DataLimitError	データリミットエラー 指定されたファイルに格納されているデータ数が上限を超えました
UnknownGraphType	グラフタイプ不明 指定されたグラフタイプが不明です
NotSupportGraphType	未対応グラフタイプ 指定されたグラフタイプは未対応です
GraphFormatError	グラフフォーマットエラー グラフの項目設定 (ページ・グラフ領域、軸、点、棒設定) に不正な設定があります
NotSupportDrawError	描画未対応 未対応の描画を実行しました
DrawFormatError	描画形式エラー 対応していない描画を実行しました
DrawProcLimitError	描画同時実行数エラー 同時描画数の上限を超えました
DrawInternalError	描画クラス内部エラー 描画モジュール内部エラーが発生しました
ProcAbort	処理中断 正常に処理が中断されました
ProcLimitOver	同時処理数オーバー 同時実行コマンド数の上限を超えました
InternalError	内部エラー GraphWeaver 内部エラーが発生しました

GraphWeaver for Visio

ColorIndex: 色

メンバ名	内 容
None	未定義(使用不可)
Transparence	透明
White	白
Black	黒
Red	赤
Green	緑
Blue	青
Yellow	黄
Magenta	マゼンタ
Cyan	シアン
Brown	茶