

# eDocuTool.net<sup>®</sup>

## 利用マニュアル

### 帳票印刷制御DLL

OOXML (ISO/IEC 規格対応)

### Excel2007 版



株式会社 **R&Dソフトウェア**

〒454-0012 名古屋市中川区尾頭橋四丁目13番7号 nabi金山 401  
Tel. 052-331-3871 URL <http://www.rndsoftware.net>

# eDocuTool.net

## Excel 版の利用について

本マニュアル記載のツールはマイクロソフト社の.NET（ドットネット）開発環境化（特にASP.NETによるWEBシステム開発）に於けるプログラム開発に活用するためのものである。（VS2003, VS2005 及び VS2008 に対応している）

本書のインターフェースに基づいてVB.NET及びC#.NETによるプログラムから適宜DLLを呼び出し業務処理プログラムを開発するとプログラムステップは大幅に減少し開発工数も削減できる。また、従来の一般的なアプリケーションプログラムの開発には、項目設定数の多い印刷制御の計表の設計とそのコーディングに多大な労力を要した。本ツールはこのシステム開発上の印刷制御に係る諸問題を一気に解消し、本ツールを利用することでプログラムステップ数が大幅に減少する。また、システム開発途上で作業負担が多い、帳票上のフォーマットの項目位置、項目内容の調整等についても追加・修正・変更が容易となる。なお、この帳票開発ツールのベースとなる、新規格「OOXML」は、2008年03月31日にISO/IEC規格として承認されたが、弊社ではこの「OOXML」について、Microsoft社のOffice2007のXML構造を一早く分析し、今般本規格に対応した文書/計表作成ツールとして完成させた。

平成20年9月

eDocuTool.net Reporting Solution

# eDocuTool.net

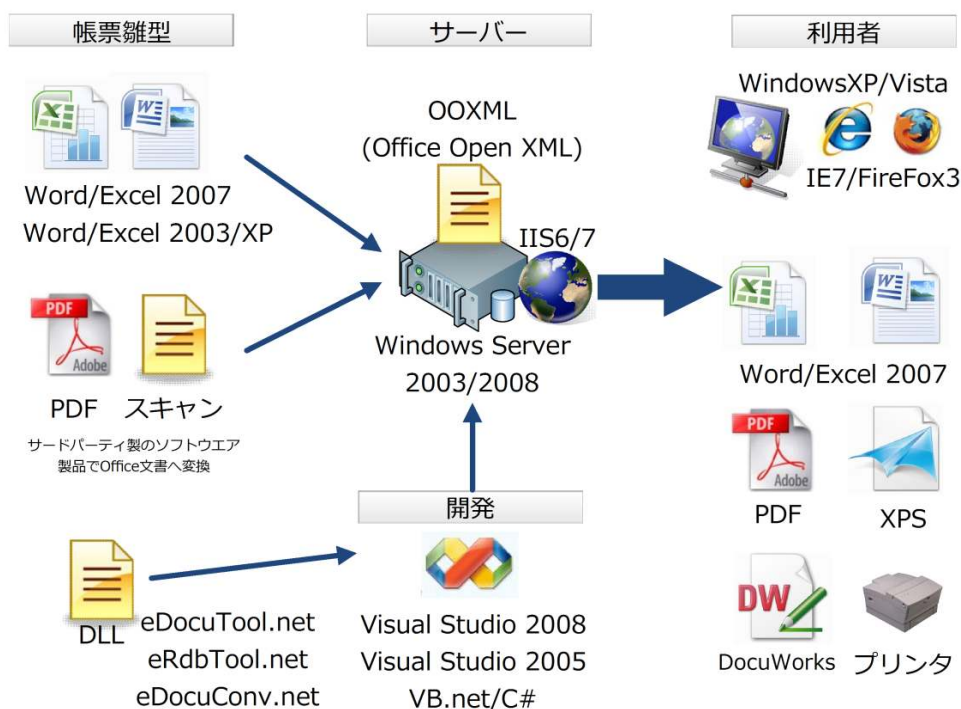
自社システム開発で業務効率化とコスト削減

短期間でタイムリーに効率的で効果的なシステム開発を行う  
**eDocuTool.netを使用したシステム開発**

「特にWebシステムにおける帳票システムを短期間で効率的に開発したい」  
このようなニーズにお応えする.NET Framework対応の帳票システム開発ツールです

WEB帳票システム開発のイメージ

Word/Excelを利用して帳票雛型を作成し、VisualStudioでプログラム開発を行います  
サーバーのIISへWEBアプリケーションを展開し、利用者はブラウザからアプリを実行します



# 目次

項番	初期処理DLL名	機能と概要	頁
	はじめに	eDocuTool.net の Excel バージョンについて	1
1.	startPage	ページ制御及びライン制御のページ開始処理	4
2.	SelectSheet	ページ制御のシート開始処理	5
3.	SelectSheetWithWriteLine	ライン制御のシート開始処理	6
項番	印刷領域追加処理DLL名	機能と概要	頁
4.	setAddPage	ページ制御のページ追加	7
5.	setAddLine	ライン制御の一行（複数行含む）のデータ域を追加	8
6.	setAddNullLine	ライン制御の一行（複数行含む）の空行を追加	9
7.	setHeader	ライン制御のヘッダーを追加	10
8.	setFooter	ライン制御のフッターを追加	11
項番	データ設定処理DLL名	機能と概要	頁
9.	setFieldData	Excel 計表の定義フィールドにデータを転送	12
10.	setFieldDataAll	Excel 計表の同定義フィールド全てにデータを転送	14
11.	setFieldDataWithAttribute	Excel 計表の定義フィールドにその表示項目の文字列の色、セルの背景色を定義し併せて印字データを設定する	16
12.	setFieldDataWithAttributeAll	Excel 計表の同定義フィールド全てにその表示項目の文字列の色、セルの背景色を定義し併せて印字データを設定する	17
13.	setTextData	Excel 計表の定義フィールドにテキストを転送（ストリング処理）	18
14.	setTextDataAll	Excel 計表の同定義フィールド全てにテキストを転送（ストリング処理）	19
15.	setShapeTextData	図形の定義フィールドにテキストを転送（ストリング処理）	20
16.	setSmartArtTextData	SmartArt の定義フィールドにテキストを転送（ストリング処理）	21
17.	setWordArtTextData	ワードアートの定義フィールドにテキストを転送（ストリング処理）	22
18.	setFieldExpression	Excel 計表の定義フィールドに数式等を設定	23
19.	setLinkURLData	ハイパーリンクの定義フィールドにアドレスデータを設定する	24
20.	setHeaderFooterTextData	ヘッダー/フッターの定義フィールドにテキストを転送	25
21.	setFieldDataClear	Excel 計表の残存定義フィールドを強制的にクリアする	26
22.	setRowHidden	Excel 計表の残存定義フィールド先頭行を非表示にする	27
23.	setRowHiddenAll	Excel 計表の残存定義フィールド行全てを非表示にする	28
24.	setPicData	画像の置換	29
25.	setPicDataAll	同定義画像の一括置換	30
26.	delPicData	画像の消去	31
項番	終了処理系DLL名	機能と概要	頁
27.	saveSheet	Excel シートの終了処理	32
28.	endPage	Excel 計表の終了処理	33
項番	印刷処理系DLL名	機能と概要	頁
29.	xlsxToPdf	XLSX ファイルから PDF ファイルへ変換	34
30.	xlsxToXps	XLSX ファイルから XPS ファイルへ変換	35
31.	xlsxToXls	XLSX ファイルから XLS ファイル（旧 Excel 形式）へ変換	36
32.	xlsxToPrinter	XLSX ファイルからプリンタ出力	37

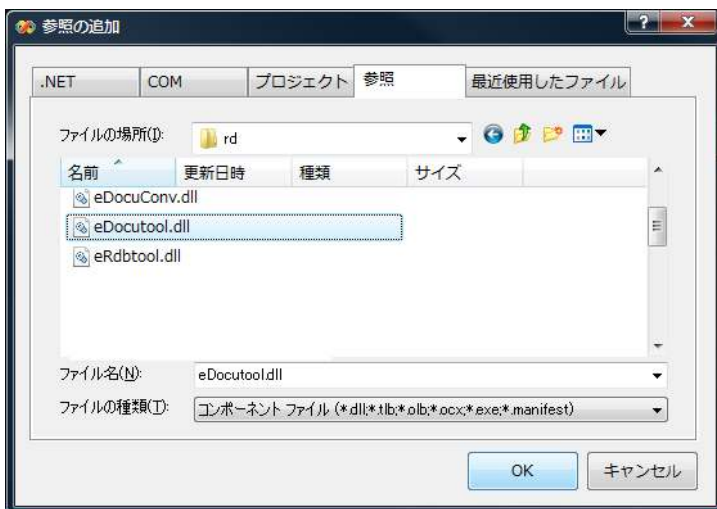
項番	データバインドDLL名	機能と概要	頁
33.	ページバージョンデータバインド		38
	eDocutoolDataBinder 定義	データバインド処理の事前設定	38
	eDocutoolDataBinder データバインド	データバインド	38
	eDocutoolDataBinder 結果取得	データバインド後の結果取得方法	39
	eDocutoolDataBinder 記述例と解説	記述例と解説	40
34.	ラインバージョンデータバインド		41
	eDocutoolDataBinder 定義	データバインド処理の事前設定	41
	eDocutoolDataBinder データバインド	データバインド	41
	eDocutoolDataBinder 結果取得	データバインド後の結果取得方法	41
	eDocutoolDataBinder 記述例と解説	記述例と解説	42
項番	本 DLL の利用事例	機能と概要	頁
資料	本ツールの利用方法	印刷制御に関する作業イメージ	資料

# はじめに

## eDocuTool.net の Excel バージョンについて

eDocuTool.net の Excel バージョンを利用するにあたって、開発環境の準備を行う必要があります。以下の設定が完了している前提で各関数の解説を行います。

### ① 参照の追加で DLL を登録する (VisualStudio2008 の場合の参照設定)



### ② オブジェクトの定義 (eDocuTool.net)

(例) VB.net の場合

```
'Imports で定義
Imports RDSOft

'eDocuTool.net オブジェクトの作成
Dim edoc = New eDocuTool.Excel
```

(例) C# の場合

```
//using で定義
using RDSOft;

// eDocuTool.net オブジェクトの作成
eDocuTool.Excel edoc = new eDocuTool.Excel ();
```

### ③オブジェクトの定義 (moconv)

(例) VB.net の場合

```
'moconv オブジェクトの作成  
Dim moc = New moconv
```

(例) C#の場合

```
// moconv オブジェクトの作成  
moconv moc = new moconv ();
```

### ④オブジェクトの定義 (eDocuToolDataBinder)

(例) VB.net の場合

```
'eDocuToolDataBinder オブジェクトの作成 (ページバージョン)  
Dim edocBind = New eDocuToolDataBinder.Excel.PageControl  
'eDocuToolDataBinder オブジェクトの作成 (ラインバージョン)  
Dim edocBind = New eDocuToolDataBinder.Excel.LineControl
```

(例) C#の場合

```
//eDocuToolDataBinder オブジェクトの作成 (ページバージョン)  
eDocuToolDataBinder.Excel.PageControl edocBinder = new eDocuToolDataBinder.Excel.PageControl ();  
//eDocuToolDataBinder オブジェクトの作成 (ラインバージョン)  
eDocuToolDataBinder.Excel.LineControl edocBinder = new eDocuToolDataBinder.Excel.LineControl ();
```

### ⑤eDocuTool.net の処理の流れ

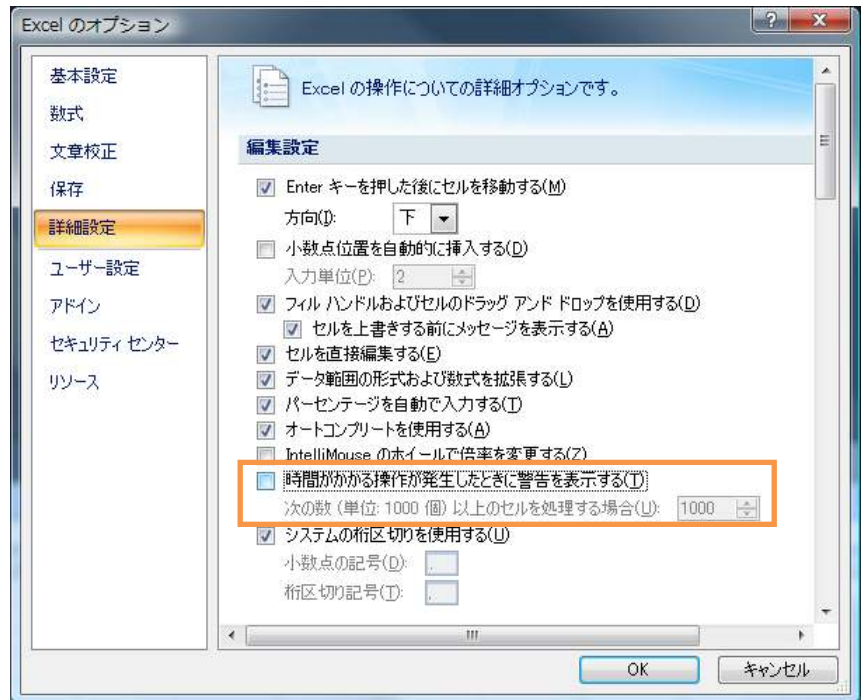
ページバージョン・ラインバージョン

① ページ読み込み	startPage("ファイル名");
② シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④ レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");
⑥ シートの終了	saveSheet ()
⑦ ページをクローズ	endPage("ファイル名");

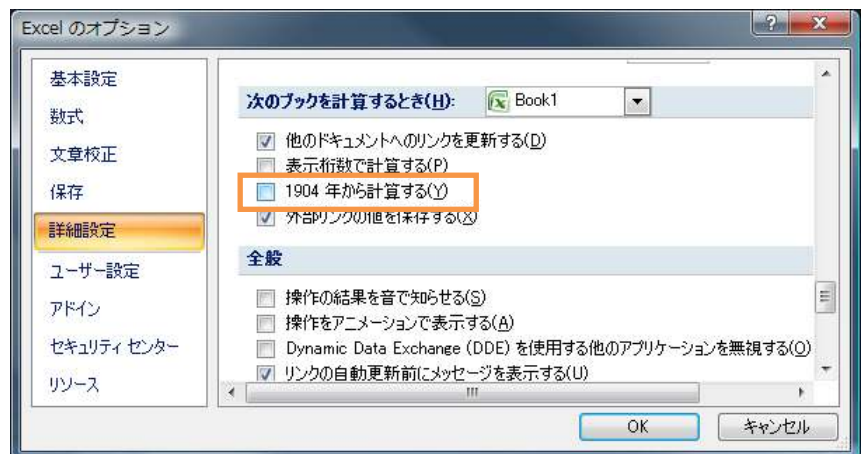
## eDocuTool.net を利用する際の Excel 帳票設計について

- ・データ量の多い帳票を作成する場合の設定  
(Excel ファイル全体に対する設定)

- ①Excel を開く
- ②Office ボタンをクリックする
- ③Excel のオプションボタンをクリックする
- ④詳細設定を選択する
- ⑤「編集設定」の「時間がかかる操作が発生したときに警告を表示する」のチェックを外す



※Excel 帳票を設計する際「次のブックを計算するとき」の「1904 年から計算する」のチェックを入れると日付が正しく表示されない為チェックは入れないでください



# 1. startPage

## Excel 計表出力の基本となる定義ファイルの読み込み 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

edoc.startPage ( " Excel 計表定義ファイル名 " )

例 ファイル名が " A B C " の場合

edoc.startPage ( " A B C " )

V B 例

edoc.startPage ( " A B C " )

C # 例

edoc.startPage ( " A B C " );



# 2 . selectSheet

## Excel シートの創生と開始

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	<b>selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")</b>
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.selectSheet( " シート名 " )

例 シート名が " 名古屋 " の場合

```
edoc.selectSheet ( " 名古屋 " )
```

VB 例

```
edoc.selectSheet ( " 名古屋 " )
```

C # 例

```
edoc.selectSheet ( " 名古屋 " );
```

# 3 . SelectSheetWithWriteLine

## ラインバージョンのExcelシートの創生と開始

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / <b>SelectSheetWithWriteLine ("シート名")</b>
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.selectSheetWithWriteLine (シート名, 先頭行番号, 末尾行番号)

例 シート名が " 名古屋 "、先頭行番号が5、末尾行番号が15の場合

edoc.selectSheetWithWriteLine ( " 名古屋 " , 5 , 15 )

#### VB例

edoc.selectSheetWithWriteLine ( " 名古屋 " , 5 , 15 )

#### C #例

edoc.selectSheetWithWriteLine ( " 名古屋 " , 5 , 15 );

# 4 . setAddPage

## Excel 計表の定義フィールドを一頁追加

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	<b>setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter</b>
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.setAddPage ()

#### 記述例

```
edoc.setAddPage ()
```

#### V B 例

```
edoc.setAddPage ()
```

#### C # 例

```
edoc.setAddPage ();
```

※ 2回目以降の setAddPage は改ページを行う

# 5 . setAddLine

ラインバージョンの Excel 計表の定義フィールドのボディ  
(データーテーブル部分) に複数行のデータ域を追加

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ <b>setAddLine</b> / setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

edoc.setAddLine ()

### 記述例

```
edoc.setAddLine ()
```

### V B 例

```
edoc.setAddLine ()
```

### C # 例

```
edoc.setAddLine ();
```

# 6 . setAddNullLine

ラインバージョンの Excel 計表の定義フィールドのボディ  
(データーテール部分)に一行（複数行含む）の空行を追加

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ <b>setAddNullLine</b> / setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

edoc.setAddNullLine ()

### 記述例

```
edoc.setAddNullLine ()
```

### V B 例

```
edoc.setAddNullLine ()
```

### C # 例

```
edoc.setAddNullLine () ;
```

# 7. setHeader

## Excel 計表の定義フィールドにヘッターを追加

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ <b>setHeader</b> / setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.setHeader ()

#### 記述例

```
edoc.setHeader ()
```

#### V B 例

```
edoc.setHeader ()
```

#### C # 例

```
edoc.setHeader ();
```

# 8 . setFooter

## Excel 計表の定義フィールドにフッターを追加

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ <b>setFooter</b>
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.setFooter ()

#### 記述例

```
edoc.setFooter ()
```

#### V B 例

```
edoc.setFooter ()
```

#### C # 例

```
edoc.setFooter () ;
```

# 9. setFieldData

## Excel 計表の定義フィールドにデータを転送

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setFieldData/ setFieldDataWithAttribute</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.setFieldData ( " 出力定義項目 " , " 出力項目 (文字列型・数値型・日付型) " )

例 (文字列型)    出力定義項目である " string " に " 010 " と出力する場合

```
edoc.setFieldData ( " string " , " 010 " )
```

エクセルで作成した定義計表上に " string " と定義した欄に " 010 " が設定される。  
複数個同名がある場合は計表左上方から順次設定される。

VB例

```
edoc.setFieldData ( " 出力定義項目 " , " 出力項目 (文字列型) " )
```

C #例

```
edoc.setFieldData ( " 出力定義項目 " , " 出力項目 (文字列型) " );
```



例（数値型） 出力定義項目である " decimal " に " 100 " と出力する場合

```
edoc.setFieldData ( " decimal " , 100 )
```

エクセルで作成した定義計表上に " decimal " と定義した欄に " 100 " が設定される。  
複数個同名がある場合は計表左上方から順次設定される。

V B 例

```
edoc.setFieldData ( " 出力定義項目 " , 出力項目 (数値型) )
```

C # 例

```
edoc.setFieldData ( " 出力定義項目 " , 出力項目 (数値型) );
```

例（日付型） 出力定義項目である " datetime " に " 現在の日時 " を出力する場合

```
edoc.setFieldData ( " datetime " , DateTime.Now)
```

エクセルで作成した定義計表上に " datetime " と定義した欄に " 現在の日時 " が設定される。  
※日付型で出力する際、出力定義した欄には必ずセルの書式設定[表示形式]で日付または時刻を設定すること。  
複数個同名がある場合は計表左上方から順次設定される。

V B 例

```
edoc.setFieldData ( " 出力定義項目 " , 出力項目 (日付型) )
```

C # 例

```
edoc.setFieldData ( " 出力定義項目 " , 出力項目 (日付型) );
```

**文字列型・数値型の説明** ※出力項目が文字列型・数値型によって出力結果が変わる場合がある

例 出力項目に " 0 " が含まれている場合

出力項目	出力結果 (文字列型)	出力結果 (数値型)
010	010	10
001	001	1
100	100	100

# 10. setFieldDataAll

Excel 計表の同定義フィールド全てにデータを転送

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setFieldDataAll/ setFieldDataWithAttributeAll</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

edoc.setFieldDataAll ( " 出力定義項目 " , " 出力項目 (文字列型・数値型・日付型) " )

例 (文字列型) 出力定義項目である " string " 全てに " 010 " と出力する場合

edoc.setFieldDataAll ( " string " , " 010 " )

エクセルで作成した定義計表上に " string " と定義した欄全てに " 010 " が設定される。

VB例

edoc.setFieldDataAll ( " 出力定義項目 " , " 出力項目 (文字列型) " )

C # 例

edoc.setFieldDataAll ( " 出力定義項目 " , " 出力項目 (文字列型) " );

例（数値型） 出力定義項目である " decimal " 全てに " 100 " と出力する場合

```
edoc.setFieldDataAll ( " decimal " , 100 )
```

エクセルで作成した定義計表上に " decimal " と定義した欄全てに " 100 " が設定される。

V B 例

```
edoc.setFieldDataAll ( " 出力定義項目 " , 出力項目 (数値型) )
```

C # 例

```
edoc.setFieldDataAll ( " 出力定義項目 " , 出力項目 (数値型) );
```

例（日付型） 出力定義項目である " datetime " に " 現在の日時 " を出力する場合

```
edoc.setFieldData ( " datetime " , DateTime.Now)
```

エクセルで作成した定義計表上に " datetime " と定義した欄全てに " 現在の日時 " が設定される。

※日付型で出力する際、出力定義した欄には必ずセルの書式設定[表示形式]で日付または時刻を設定すること。

V B 例

```
edoc.setFieldDataAll ( " 出力定義項目 " , 出力項目 (日付型) )
```

C # 例

```
edoc.setFieldDataAll ( " 出力定義項目 " , 出力項目 (日付型) );
```

**文字列型・数値型の説明** ※出力項目が文字列型・数値型によって出力結果が変わる場合がある

例 出力項目に " 0 " が含まれている場合

出力項目	出力結果 (文字列型)	出力結果 (数値型)
010	010	10
001	001	1
100	100	100

# 11. setFieldDataWithAttribute

Excel 計表の定義フィールドにその表示項目の文字列の色、セルの背景色を定義し、併せて印字データを設定

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ <b>setFieldDataWithAttribute</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

```
edoc.setFieldDataWithAttribute ( " 出力定義項目 " , " 出力項目 " ,  
                                " Attribute 1 " , " Attribute 2 " )  
                                フォントカラー      セルの背景色
```

カラーはRed, Blue, White, Black, Green, Yellow等又はカラー指定RGB 9桁で指定する

例 出力定義項目である " H E A D E R " に " 東京 " を表示、フォントカラー：白、背景色：青の場合

```
edoc.setFieldDataWithAttribute ( " H E A D E R " , " 東京 " , Color.White, Color.Blue)
```

V B 例

```
edoc.setFieldDataWithAttribute ( " 出力定義項目 " , " 出力項目 " , " Attribute 1 " , " Attribute 2 " )
```

C # 例

```
edoc.setFieldDataWithAttribute ( " 出力定義項目 " , " 出力項目 " , " Attribute 1 " , " Attribute 2 " );
```

## アトリビュート一覧

Attribute 1	Attribute 2
フォントカラー	セルの背景色
Color.Empty : 変更なし	Color.Empty : 変更なし
Green等を指定又はRGB 9桁で指定 例 白 Color.FromArgb(255, 255, 255)	Green等を指定又はRGB 9桁で指定 例 青 Color.FromArgb(0, 0, 255)

# 12. setFieldDataWithAttributeAll

Excel 計表の同定義フィールド全てにその表示項目の文字列の色、セルの背景色を定義し、併せて印字データを設定

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setFieldDataAll/ setFieldDataWithAttributeAll</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

```
edoc.setFieldDataWithAttributeAll ( " 出力定義項目 " , " 出力項目 " ,  
                                     " Attribute 1 " , " Attribute 2 " )  
                                     フォントカラー      セルの背景色
```

カラーはRed, Blue, White, Black, Green, Yellow等又はカラー指定RGB 9桁で指定する

例 出力定義項目である " H E A D E R " 全てに " 東京 " を表示、フォントカラー：白、背景色：青の場合

```
edoc.setFieldDataWithAttributeAll ( " H E A D E R " , " 東京 " , Color.White, Color.Blue)
```

## V B 例

```
edoc.setFieldDataWithAttributeAll ( " 出力定義項目 " , " 出力項目 " , " Attribute 1 " , " Attribute 2 " )
```

## C # 例

```
edoc.setFieldDataWithAttributeAll ( " 出力定義項目 " , " 出力項目 " , " Attribute 1 " , " Attribute 2 " );
```

## アトリビュート一覧

Attribute 1	Attribute 2
フォントカラー	セルの背景色
Color.Empty : 変更なし	Color.Empty : 変更なし
Green等を指定又はRGB 9桁で指定 例 白 Color.FromArgb(255, 255, 255)	Green等を指定又はRGB 9桁で指定 例 青 Color.FromArgb(0, 0, 255)

# 13. setTextData

Excel 計表の定義フィールドにテキストを転送（文字列を追加挿入する機能）

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setTextData</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setTextData ( " 出力定義項目 " , " 出力項目 " )`

例 出力定義項目である " 項目 1 " に " 東京～名古屋 " と出力する場合

```
edoc.setTextData ( " 項目 1 " , " 東京～名古屋 " )
```

エクセルで作成した文字列の中に定義名標を " 項目 1 " と定義した場合 " 東京～名古屋 " が設定される。  
複数個同名がある場合は計表左上方から順次設定される。本DLLはストリング処理を行う機能である。

### VB例

```
edoc.setTextData ( " 出力定義項目 " , " 出力項目 " )
```

### C#例

```
edoc.setTextData ( " 出力定義項目 " , " 出力項目 " );
```

# 14. setTextDataAll

Excel 計表の同定義フィールド全てにテキストを転送（文字列を追加挿入する機能）

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setTextDataAll</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setTextDataAll ( " 出力定義項目 " , " 出力項目 " )`

例 出力定義項目である " 項目 1 " 全てに " 東京～名古屋 " と出力する場合

```
edoc.setTextDataAll ( " 項目 1 " , " 東京～名古屋 " )
```

エクセルで作成した文字列の中に定義名標を " 項目 1 " と定義した全てに " 東京～名古屋 " が設定される。  
本DLLはストリング処理を行う機能である。

### VB例

```
edoc.setTextDataAll ( " 出力定義項目 " , " 出力項目 " )
```

### C #例

```
edoc.setTextDataAll ( " 出力定義項目 " , " 出力項目 " );
```

# 15. setShapeTextData

## 図形の定義フィールドにテキストを転送

### 処理の流れ

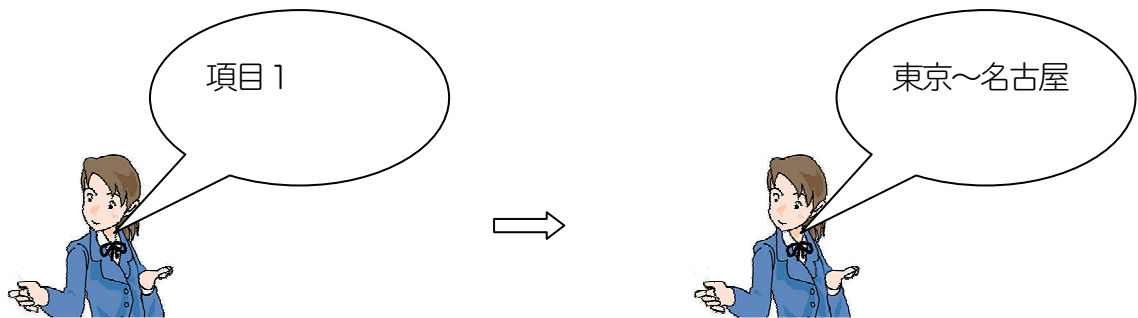
①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setShapeTextData</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

`edoc.setShapeTextData ( " 出力定義項目 " , " 出力項目 " )`

例 出力定義項目である " 項目 1 " に " 東京～名古屋 " と出力する場合

`edoc.setShapeTextData ( " 項目 1 " , " 東京～名古屋 " )`



エクセルで作成した図形の中に定義名標を " 項目 1 " と定義した場合 " 東京～名古屋 " が設定される。  
本DLLはストリング処理を行う機能である。

### V B 例

`edoc.setShapeTextData ( " 出力定義項目 " , " 出力項目 " )`

### C # 例

`edoc.setShapeTextData ( " 出力定義項目 " , " 出力項目 " )`



# 16. setSmartArtTextData

## SmartArt の定義フィールドにテキストを転送

### 処理の流れ

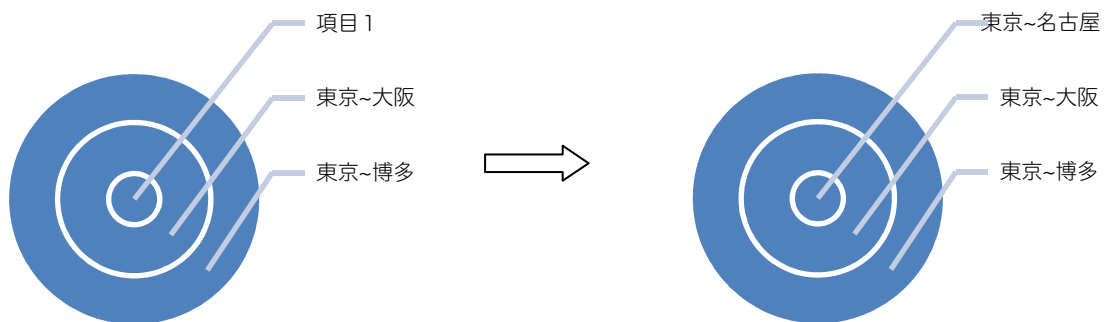
①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setSmartArtTextData</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

`edoc.setSmartArtTextData ( " 出力定義項目 " , " 出力項目 " )`

例 出力定義項目である " 項目 1 " に " 東京～名古屋 " と出力する場合

`edoc.setSmartArtTextData ( " 項目 1 " , " 東京～名古屋 " )`



エクセルで作成したSmartArtの中に定義名標を " 項目 1 " と定義した場合 " 東京～名古屋 " が設定される。  
本DLLはストリング処理を行う機能である。

### V B 例

`edoc.setSmartArtTextData ( " 出力定義項目 " , " 出力項目 " )`

### C # 例

`edoc.setSmartArtTextData ( " 出力定義項目 " , " 出力項目 " );`

# 17. setWordArtTextData

ワードアートの定義フィールドにテキストを転送

## 処理の流れ

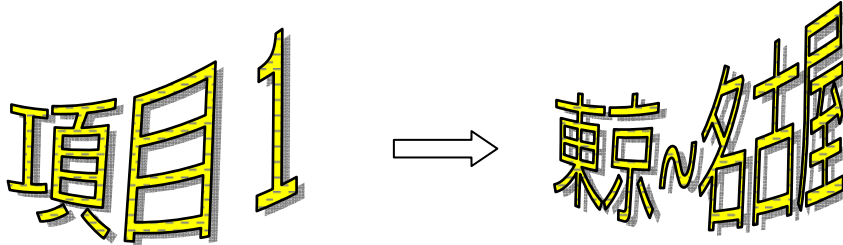
①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setWordArtTextData
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

edoc.setWordArtTextData ( " 出力定義項目 " , " 出力項目 " )

例 出力定義項目である " 項目 1 " に " 東京～名古屋 " と出力する場合

edoc.setWardArtTextData ( " 項目 1 " , " 東京～名古屋 " )



エクセルで作成したワードアートの中に定義名標を " 項目 1 " と定義した場合 " 東京～名古屋 " が設定される。  
本DLLはストリング処理を行う機能である。

## V B 例

```
edoc.setWordArtTextData ( " 出力定義項目 " , " 出力項目 " )
```

## C # 例

```
edoc.setWordArtTextData ( " 出力定義項目 " , " 出力項目 " );
```

# 18. setFieldExpression

Excel 計表の定義フィールドに数式等を設定

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名にEXCEL関数 (任意の組み込みが可能) を設定	setFieldExpression
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setFieldExpression ( " 出力定義項目 " , " EXCEL 関数表記 " )`

例 集計項目にEXCELの関数文字列を転送する場合

```
edoc.setFieldExpression ( " Fxarea " , " EXCEL関数表記の文字列 " )
```

V B 例

```
edoc.setFieldExpression ( " 出力定義項目 " , " EXCEL関数表記の文字列 " )
```

C # 例

```
edoc.setFieldExpression ( " 出力定義項目 " , " EXCEL関数表記の文字列 " );
```

# 19. setLinkURLData

ハイパーリンクの定義フィールドにアドレスデータを設定する

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内のハイパーリンクにアドレスを設定	<b>setLinkURLData</b>
⑤ページ内の項目名全てクリア	delPicData
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

**edoc.setLinkURLData ( " 出力定義項目 " , "リンク先のアドレス" )**

例 出力定義項目の " URL " にアドレスを設定する場合

**edoc.setLinkURLData ( " URL " , "リンク先のアドレス" )**

出力定義項目はハイパーリンクの挿入のヒント設定：ヒントテキストで確認。

VB例

**edoc.setLinkURLData ( " URL " , "リンク先のアドレス" )**

C #例

**edoc.setLinkURLData ( " URL " , "リンク先のアドレス" );**

# 20. setHeaderFooterTextData

ヘッダー/フッターの定義フィールドにテキストを転送

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④ヘッダー/フッター内の項目名に値を設定	<b>setHeaderFooterTextData</b>
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setHeaderFooterTextData ( " 出力定義項目 " , " 出力項目 " )`

例 出力定義項目である " Header " に " 名古屋 " と出力する場合

`edoc.setHeaderFooterTextData ( " Header " , " 名古屋 " )`

ページ設定のヘッダー/フッターで設定した文字列の中に定義名標を " Header " と定義した場合 " 名古屋 " が設定される。

### V B 例

`edoc.setHeaderFooterTextData ( " 出力定義項目 " , " 出力項目 " )`

### C # 例

`edoc.setHeaderFooterTextData ( " 出力定義項目 " , " 出力項目 " );`

# 21. setFieldDataClear

定義した E x c e l 帳票の残存定義フィールドを強制的にクリアする  
(ページエンド、帳票エンドの処理を自動化)

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

edoc.setFieldDataClear( " 出力定義項目 " )

例 ページエンド処理で出力定義項目である残存 " ListArea " をクリアする場合

```
edoc.setFieldDataClear ( " ListArea " )
```

この場合 " L i s t A r e a " とエクセル帳票で作成した全ての同名定義欄がクリアされる。

V B 例

```
edoc.setFieldDataClear ( " 出力定義項目 " )
```

C # 例

```
edoc.setFieldDataClear ( " 出力定義項目 " );
```

# 22. setRowHidden

定義した E x c e l 帳票の残存定義フィールド行を非表示にする  
(ページエンド、帳票エンドの処理を自動化)

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setRowHidden( " 出力定義項目 " )`

例 ページエンド処理で出力定義項目である残存 " ListArea " の先頭行を非表示にする場合

```
edoc.RowHidden( " ListArea " )
```

この場合 " L i s t A r e a " とエクセル帳票で作成した同名定義欄の先頭行が非表示にされる。

V B 例

```
edoc.setRowHidden ( " 出力定義項目 " )
```

C # 例

```
edoc.setRowHidden ( " 出力定義項目 " );
```

# 23. setRowHiddenAll

定義した E x c e l 帳票の残存定義フィールド行を非表示にする  
(ページエンド、帳票エンドの処理を自動化)

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHiddenAll("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setRowHiddenAll( " 出力定義項目 " )`

例 ページエンド処理で出力定義項目である残存 " ListArea " の行を全て非表示にする場合

```
edoc.RowHiddenAll( " ListArea " )
```

この場合 " L i s t A r e a " とエクセル帳票で作成した全ての同名定義欄の残存行が非表示にされる。

V B 例

```
edoc.setRowHiddenAll ( " 出力定義項目 " )
```

C # 例

```
edoc.setRowHiddenAll ( " 出力定義項目 " );
```



# 24. setPicData

Excel 計表の定義フィールドに画像データを設定する

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setPicData</b>
⑤ページ内の項目名全てクリア	delPicData
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setPicData ( " 出力定義項目 " , @"画像データの絶対パス")`

例 出力定義項目の " ListArea " に画像領域を設定する場合

`edoc.setPicData ( " ListArea " , @"画像データの絶対パス")`

出力定義項目は画像のサイズとプロパティで確認。

VB 例

`edoc.setPicData ( " ListArea " , @"画像データの絶対パス")`

C # 例

`edoc.setPicData ( " ListArea " , @"画像データの絶対パス") ;`

# 25. setPicDataAll

Excel 計表の同定義フィールド全てに画像データを設定する

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	<b>setPicDataAll</b>
⑤ページ内の項目名全てクリア	delPicData
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

`edoc.setPicDataAll ( " 出力定義項目 " , @"画像データの絶対パス")`

例 出力定義項目の " ListArea " 全てに画像領域を設定する場合

`edoc.setPicDataAll ( " ListArea " , @"画像データの絶対パス")`

出力定義項目は画像のサイズとプロパティで確認。

VB例

`edoc.setPicDataAll ( " ListArea " , @"画像データの絶対パス")`

C #例

`edoc.setPicDataAll ( " ListArea " , @"画像データの絶対パス") ;`

# 26.delPicData

Excel 計表の定義フィールドに画像データをクリアする

## 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setPicData
⑤ページ内の項目名全てクリア	delPicData
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

## 書式

edoc.delPicData ( " 出力定義項目 " )

例 出力定義項目の " ListArea " の画像領域をクリアする場合

```
edoc.delPicData ( " ListArea " )
```

出力定義項目は画像のサイズとプロパティで確認。

この場合 " L i s t A r e a " とエクセル計表で作成した全ての同名定義画像がクリアされる。

V B 例

```
edoc.delPicData ( " ListArea " )
```

C # 例

```
edoc.delPicData ( " ListArea " );
```

# 27. saveSheet

## E x c e l 計表のシート終了処理

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.saveSheet ()

例 E x c e l 計表の該当シートに全ての出力データ処理が終了した時点で本DLLを呼び出す

```
edoc.saveSheet ()
```

V B 例

```
edoc.saveSheet ()
```

C # 例

```
edoc.saveSheet ();
```

# 28. endPage

## Excel 計表出力の終了処理

### 処理の流れ

①ページ読み込み	startPage("ファイル名");
②シートの開始	selectSheet ("シート名") / SelectSheetWithWriteLine ("シート名")
③ページまたはラインの追加	setAddPage/ setAddLine/ setAddNullLine/ setHeader/ setFooter
④レポート内の項目名に値を設定	setFieldData/ setFieldDataWithAttribute
⑤ページ内の項目名全てクリアまたは非表示	setFieldDataClear("対象項目名");/ setRowHidden("対象項目名");
⑥シートの終了	saveSheet ()
⑦ページをクローズ	endPage("ファイル名");

### 書式

edoc.endPage(Excel 計表定義ファイル名)

例 Excel 計表に全ての出力データが処理完了となった時点で本DLLを呼び出す

ファイル名を "AAA" とする場合

```
edoc.endPage ( "AAA" )
```

VB例

```
edoc.endPage ( "AAA" )
```

C#例

```
edoc.endPage ( "AAA" );
```

# 29.xlsxToPdf

## XLSXファイルからPDFファイルへ変換

### 書式

```
moc.xlsxToPdf(string inputFileName, string outputPdfFilename,  
               string outputPdfQuality, int outputPdfStartPage,  
               int outputPdfEndPage, string[] outputPdfSheets,  
               out string result, out string resultMessage)
```

inputFileName : XLSXファイルの絶対パス

outputPdfFilename : PDFファイルを作成する場合の絶対パス

outputPdfQuality : PDFファイルの出力精度を指定 ( 「xlQualityStandard」 : 高画質 「xlQualityMinimum」 : 低画質 )

outputPdfStartPage : 出力開始ページ

outputPdfEndPage : 出力終了ページ

※ページ数を指定しない場合 (全頁出力) は、開始、終了ともに0をセット

outputPdfSheets : 印刷範囲 (現在のバージョンは未対応: null をセット)

result : 結果 ( 「1」 : 成功 「9」 : 失敗 )

resultMessage : resultが「9」の場合のメッセージ

### V B例

```
moc.xlsxToPdf( "XLSXファイル名", "PDFファイル名", "xlQualityStandard", 0, 0, null,  
               out result, out resultmessage)
```

### C #例

```
moc.xlsxToPdf( "XLSXファイル名", "PDFファイル名", "xlQualityStandard", 0, 0, null,  
               out result, out resultmessage);
```

# 30.xlsxToXps

## XLSXファイルからXPSファイルへ変換

### 書式

```
moc.xlsxToXps(string inputFileName, string outputXpsFilename,  
               string outputXpsQuality, int outputXpsStartPage,  
               int outputXpsEndPage, string[] outputXpsSheets,  
               out string result, out string resultMessage)
```

inputFileName : XLSXファイルの絶対パス

outputXpsFilename : XPSファイルを作成する場合の絶対パス

outputXpsQuality : XPSファイルの出力精度を指定 (「xlQualityStandard」 : 高画質 「xlQualityMinimum」 : 低画質)

outputXpsStartPage : 出力開始ページ

outputXpsEndPage : 出力終了ページ

※ページ数を指定しない場合 (全頁出力) は、開始、終了ともに0をセット

outputXpsSheets : 印刷範囲 (現在のバージョンは未対応 : null をセット)

result : 結果 (「1」 : 成功 「9」 : 失敗)

resultMessage : resultが「9」の場合のメッセージ

### V B例

```
moc.xlsxToXps( " XLSXファイル名 ", " XPSファイル名 ", " xlQualityStandard ", 0, 0, null,  
               out result, out resultmessage)
```

### C #例

```
moc.xlsxToXps( " XLSX ファイル名 ", " XPS ファイル名 ", " xlQualityStandard ", 0, 0, null,  
               out result, out resultmessage);
```

# 31. xlsxToXls

XLSXファイルからXLSファイル（旧Excel形式）へ変換  
書式

```
moc.xlsxToXls(string inputFileName, string outputXlsFilename,  
               out string result, out string resultMessage)
```

inputFileName : XLSXファイルの絶対パス

outputXlsFilename : XLSファイルを作成する場合の絶対パス

result : 結果 （「1」:成功 「9」:失敗 ）

resultMessage : resultが「9」の場合のメッセージ

## V B例

```
moc.xlsxToXls( " XLSXファイル名 ", " XLSファイル名 ", out result, out resultmessage)
```

## C #例

```
moc.xlsxToXls( " XLSX ファイル名 ", " XLS ファイル名 ", out result, out resultmessage);
```



# 32.xlsxToPrinter

## XLSXファイルからプリンタ出力

### 書式

```
moc.xlsxToPrinter(string inputFileName, string outputPrintername,  
                  int outputPrinterStartPage, int outputPrinterEndPage,  
                  int outputPrinterCopies, string[] outputPrinterSheets,  
                  out string result, out string resultMessage)
```

inputFileName : XLSXファイルの絶対パス

outputPrintername : 出力するプリンタ名

outputPrinterStartPage : 出力開始ページ

outputPrinterEndPage : 出力終了ページ

※ページ数を指定しない場合（全頁出力）は、開始、終了ともに0をセット

outputPrinterCopies : 出力部数

outputPrinterSheets : 印刷範囲（現在のバージョンは未対応： null をセット）

result : 結果 （「1」：成功 「9」：失敗 ）

resultMessage : resultが「9」の場合のメッセージ

### VB例

```
moc.xlsxToPrinter( " XLSXファイル名 ", " プリンタ名 ", 0, 0, 1, null, out result, out resultmessage)
```

### C #例

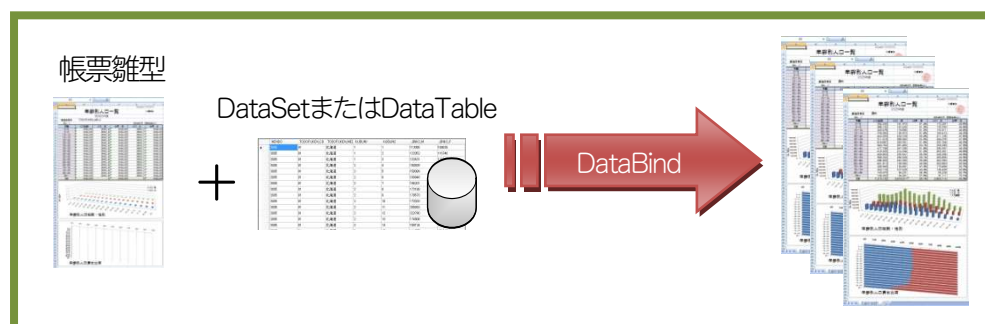
```
moc.xlsxToPrinter ( " XLSX ファイル名 ", " プリンタ名 ", 0, 0, 1, null, out result, out resultmessage);
```

# 33.eDocuToolDataBinder.Excel.PageControl

## (データバインド ページバージョン)

### 概要

帳票作成プログラム記述時における煩雑なループ処理、改ページ処理、値の設定などの命令を記述することなく、  
緒表雛型と DataTable バインドするだけで帳票作成を完結する。ステップ数の大幅な削減となる。



### 事前設定

帳票雛型の設計時に、データ設定時のラベル名をデータテーブル項目名と合わせておく必要がある。

バインド前に設定する項目名	必須	解説
sourceXlsxFileName	○	帳票雛型ファイル名
sourceXlsxSheetName	○	帳票雛型シート名
outputXlsxFileName	○	出力ファイル名
ChangingPageRowCount		改ページ設定 (1 ページあたりのレコード数)
ChangingPageConditionString		改ページ設定 (改ページ条件とする項目名)
SortConditionString		ソート設定 (ソートする項目名)
DataSource	○	データソース (DataSetまたはDataTable)
TableName		DataSet設定時のTableName指定
FieldDataBinding		Field設定でデータを帳票にセットするか (デフォルト : true)
TextDataBinding		Text設定でデータを帳票にセットするか (デフォルト : false)
headerFooterBinding		headerFooter設定でデータを帳票にセットするか (デフォルト : false)
ImageDataBinding		Image設定でデータを帳票にセットするか (デフォルト : false)
LinkURLDataBinding		LinkURL設定でデータを帳票にセットするか (デフォルト : false)
ShapeTextDataBinding		Shape設定でデータを帳票にセットするか (デフォルト : false)
SmartArtTextDataBinding		SmartArt設定でデータを帳票にセットするか (デフォルト : false)
WordArtTextDataBinding		WordArt設定でデータを帳票にセットするか (デフォルト : false)
OmissionBindingColumns		1レコード目で成功したカラムで2レコード目以降の処理を行う (デフォルト : true)
ClearRowHidden		RowHidden設定で行を非表示にするか (デフォルト : false)
バインド時に設定する項目名	必須	解説
DataBind()	○	帳票雛型とDataSourceを結合

バインド後に取得できる項目名	必須	解説
SuccessCount		データを帳票にセット成功した件数
FailureCount		データを帳票にセット失敗した件数
FieldDataBindSuccessCount		Field設定でデータを帳票にセット成功した件数
FieldDataBindFailureCount		Field設定でデータを帳票にセット失敗した件数
TextDataBindSuccessCount		Text設定でデータを帳票にセット成功した件数
TextDataBindFailureCount		Text設定でデータを帳票にセット失敗した件数
ImageDataBindSuccessCount		Image設定でデータを帳票にセット成功した件数
ImageDataBindFailureCount		Image設定でデータを帳票にセット失敗した件数
ShapeTextDataBindSuccessCount		Shape設定でデータを帳票にセット成功した件数
ShapeTextDataBindFailureCount		Shape設定でデータを帳票にセット失敗した件数
SmartArtTextDataBindSuccessCount		SmartArt設定でデータを帳票にセット成功した件数
SmartArtTextDataBindFailureCount		SmartArt設定でデータを帳票にセット失敗した件数
WordArtTextDataBindSuccessCount		WordArt設定でデータを帳票にセット成功した件数
WordArtTextDataBindFailureCount		WordArt設定でデータを帳票にセット失敗した件数
PageCount		出力ページ数
SuccessBindingColumns		Bindが成功したColumn名
FailureBindingColumns		Bindが失敗したColumn名

## 記述例 (VB.net)

```
1. Dim db_TP5Y = erdb.SqlDbGet(sqlStr, sqlc)
2. Dim eDocBinder AS new eDocuToolDataBinder.Excel.PageControl
3. eDocBinder.sourceXlsxFileName = "年齢性別一覧2"
4. eDocBinder.sourceXlsxSheetName = "年齢別統計"
5. eDocBinder.outputXlsxFileName = "年齢性別一覧_" + DateTime.Now.ToString("yyyyMMddHHmmssffffff")
6. eDocBinder.ChangingPageConditionString = "NENDO,TODOFUKEN_CD"
7. eDocBinder.SortConditionString = "NENDO,TODOFUKEN_CD,KUBUN2"
8. eDocBinder.DataSource = db_TP5Y
9. eDocBinder.DataBind()
```

## 記述例 (C#)

```
1. DataTable db_TP5Y = erdb.SqlDbGet(sqlStr, sqlc);
2. eDocuToolDataBinder.Excel.PageControl eDocBinder = new eDocuToolDataBinder.Excel.PageControl ();
3. eDocBinder.sourceXlsxFileName = "年齢性別一覧2";
4. eDocBinder.sourceXlsxSheetName = "年齢別統計";
5. eDocBinder.outputXlsxFileName = "年齢性別一覧_" + DateTime.Now.ToString("yyyyMMddHHmmssffffff");
6. eDocBinder.ChangingPageConditionString = "NENDO,TODOFUKEN_CD";
7. eDocBinder.SortConditionString = "NENDO,TODOFUKEN_CD,KUBUN2";
8. eDocBinder.DataSource = db_TP5Y;
9. eDocBinder.DataBind();
```

## 解説

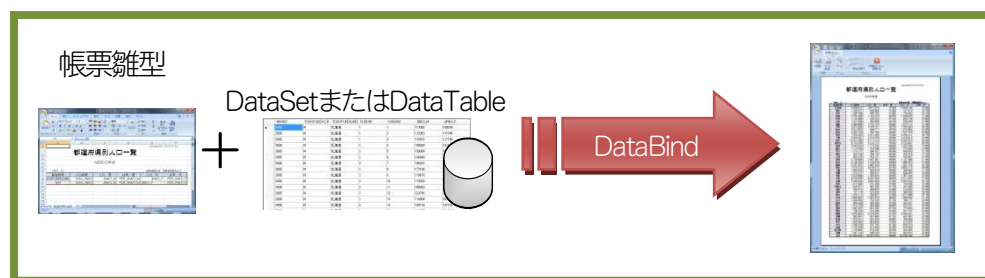
1. データテーブルをデータベースから読み込み
2. eDocuToolDataBinder オブジェクトの作成
3. 帳票雛型ファイル名の設定
4. 帳票雛型シート名の設定
5. 出力ファイル名の設定
6. 改ページ条件の設定
7. ソート条件の設定
8. データソースの設定
9. データバインド

# 34.eDocuToolDataBinder.Excel.LineControl

## (データバインド ラインバージョン)

### 概要

帳票作成プログラム記述時における煩雑なループ処理、改ページ処理、値の設定などの命令を記述することなく、緒表雛型と DataTable バインドするだけで帳票作成を完結する。ステップ数の大幅な削減となる。



### 事前設定

帳票雛型の設計時に、データ設定時のラベル名をデータテーブル項目名と合わせておく必要がある。

バインド前に設定する項目名	必須	解説
sourceXlsxFileName	○	帳票雛型ファイル名
sourceXlsxSheetName	○	帳票雛型シート名
outputXlsxFileName	○	出力ファイル名
startDetailRow	○	明細部分のレコード指定 (開始レコード) ※Excel の行番号
endDetailRow	○	明細部分のレコード指定 (終了レコード) ※Excel の行番号
SortConditionString		ソート設定 (ソートする項目名)
HeaderDataSource		ヘッダーデータソース (DataSetまたはDataTable)
DetailDataSource	○	明細データソース (DataSetまたはDataTable)
FooterDataSource		フッターデータソース (DataSetまたはDataTable)
HeaderTableName		DataSet設定時のヘッダーTableName指定
DetailTableName		DataSet設定時の明細TableName指定
FooterTableName		DataSet設定時のフッターTableName指定
FieldDataBinding		Field設定でデータを帳票にセットするか (デフォルト: true)
TextDataBinding		Text設定でデータを帳票にセットするか (デフォルト: false)
OmissionBindingColumns		1レコード目で成功したカラムで2レコード目以降の処理を行う (デフォルト: true)
バインド時に設定する項目名	必須	解説
DataBind()	○	帳票雛型とDataSourceを結合
バインド後に取得できる項目名	必須	解説
SuccessCount		データを帳票にセット成功した件数
FailureCount		データを帳票にセット失敗した件数
FieldDataBindSuccessCount		Field設定でデータを帳票にセット成功した件数
FieldDataBindFailureCount		Field設定でデータを帳票にセット失敗した件数
TextDataBindSuccessCount		Text設定でデータを帳票にセット成功した件数
TextDataBindFailureCount		Text設定でデータを帳票にセット失敗した件数
SuccessBindingColumns		Bindが成功したColumn名
FailureBindingColumns		Bindが失敗したColumn名

## 記述例 (C#)

//データベース読み込み

```
DataTable db_TP5Y = erdb.SqlDbGet(sqlStr, sqlc);
```

```
DataTable db_TP5Y_TOTAL = db_TP5Y.Clone();
```

//フッター集計行の作成

```
DataRow drF=db_TP5Y_TOTAL.NewRow();
```

```
decimal SUM_JINKO = Convert.ToDecimal(db_TP5Y.Compute("SUM(JINKO_M)+SUM(JINKO_F)", ""));
```

```
drF["JINKO_M"] = db_TP5Y.Compute("SUM(JINKO_M)", "");
```

```
drF["JINKO_F"] = db_TP5Y.Compute("SUM(JINKO_F)", "");
```

```
drF["SUM_JINKO"] = SUM_JINKO;
```

```
drF["PER_JINKO_M"] = Convert.ToDecimal(drF["JINKO_M"]) / SUM_JINKO;
```

```
drF["PER_JINKO_F"] = 1 - Convert.ToDecimal(drF["JINKO_M"]) / SUM_JINKO;
```

```
db_TP5Y_TOTAL.Rows.Add(drF);
```

//バインド

```
eDocuToolDataBinder.Excel.LineControl eDocBinder = new eDocuToolDataBinder.Excel.LineControl();
```

```
eDocBinder.sourceXlsxFileName = "都道府県別一覧";
```

```
eDocBinder.sourceXlsxSheetName = "都道府県別統計";
```

```
eDocBinder.outputXlsxFileName = "都道府県別一覧_" + DateTime.Now.ToString("yyyyMMddHHmmssffffff");
```

```
eDocBinder.startDetailRow = 8;
```

```
eDocBinder.endDetailRow = 8;
```

```
eDocBinder.HeaderDataSource = db_TP5Y;
```

```
eDocBinder.DetailDataSource = db_TP5Y;
```

```
eDocBinder.FooterDataSource = db_TP5Y_TOTAL;
```

```
eDocBinder.DataBind();
```

## 解説

1. SQL でレコード毎の集計作業を完了させておき、データベースを読み込む
2. DataTable の Compute 関数を使用し、フッター合計行の集計を行い、フッター用 DataTable へ格納
3. 帳票雛型ファイル名の設定
4. 帳票雛型シート名の設定
5. 出力ファイル名の設定
6. 明細行の位置指定 (Excel シートの 8 行目を繰り返す)
7. 1 レコード目で成功したカラムで 2 レコード目以降の処理を行う
8. データソースの設定 (ヘッダー)
9. データソースの設定 (明細)
10. データソースの設定 (フッター)
11. データバインド

## 本ツールの利用方法

1. 帳票設計を Word 文書とするか Excel 計表とするか選択する
2. Excel を選択した場合 Page 制御（1 ページ単位制御）にするか、Line 制御（1 行毎に追加制御）にするかを判断する

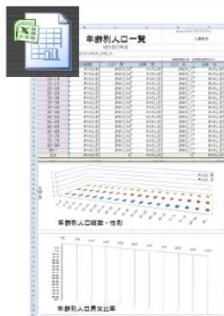
## 帳票レイアウトとラベルの設定

eDocuTool.net は MicrosoftOffice2007 に対応した帳票設計パターンを 3 種類用意

Word2007 PageControl

Excel2007 PageControl

Excel2007 LineControl



※帳票設計パターン

PageContorol はページ単位形式

LineContorol は行追加形式

3. プログラミング方式を個別記述方式（通常記述）とするか、データバインド方式（一括記述方式）にするかを定める

# 帳票プログラミング

eDocuTool.net は帳票プログラミングを2パターン用意し、記述を簡素化

通常記述

## 最小 6 命令の帳票プログラミング

[illegible]

## DataBind 記述

## DataBind()命令のみの帳票プログラミング

(帳票と DataBase の結合を行います)

```

//バインド
eDocBinder.DataSource.Excel.PaneControl.eDocBinder = new eDocBinder(DataSource.Excel.PaneControl());
eDocBinder.SourceExcelFileName = "年報別一覧" + 年報書ファイル名
eDocBinder.SourceExcelSheetName = "年報別一覧" + 年報書シート名
eDocBinder.SourceExcelFileName = "年報別一覧" + 年報書ファイル名
eDocBinder.ChangingPaneControl = 18;
eDocBinder.ChangingPaneControlString = "MENU、TOPFUNK" + 年報書オプション
eDocBinder.OmissionInitialLoading = true; //無効効率化オプション
eDocBinder.DataSource = "7992" + 年報書ポート用ID
eDocBinder.DataBind(); //データバインド

```

※サンプルプログラムは「Excel PageControl」を利用（帳票出力部分のみ抜粋）

4. データを帳票の名標（ラベル）に埋め込み、文書計表を完成させる

## 帳票の印刷制御

帳票設計パターンごとの帳票出力

Word2007 PageControl

Excel2007 PageControl

## Excel2007 LineControl



# eDocuTool.net<sup>®</sup>

## Excel 版マニュアル



株式会社 **R&Dソフトウェア**

〒454-0012 名古屋市 中川区 尾頭橋四丁目 13番7号 inabi金山 401  
Tel. 052-331-3871 URL <http://www.rndssoftware.net>