2009年9月

eDocuTool.net©

利用マニュアル

帳票印刷制御ＤＬＬ

OOXML(ISO/IEC)規格対応

Word2007 版

株式会社R&Dソフトウエア
〒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.rdsoftware.net

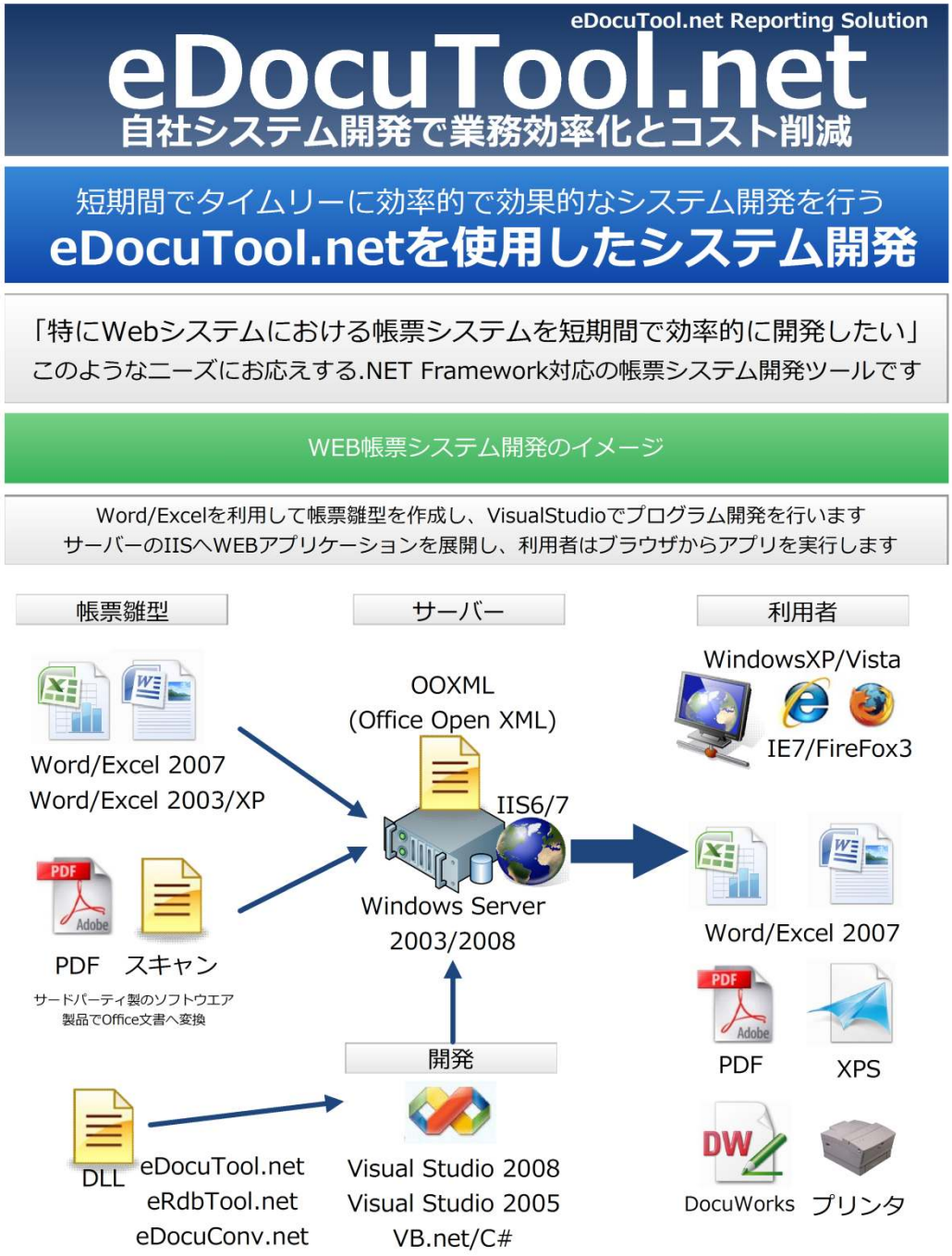
eDocuTool.net

Word 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文書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、特に Microsoft 社の Office2007 の XML 構造を昨年から一早く分析を進め、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



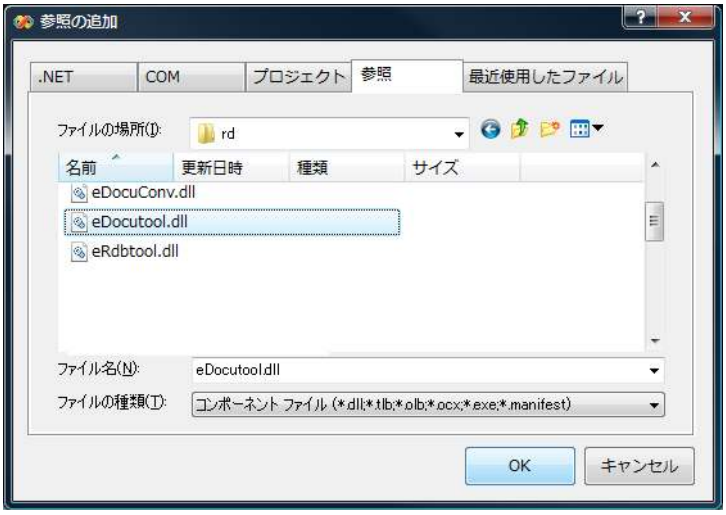
目次

はじめに

eDocuTool.net の Word バージョンについて

eDocuTool.net の Word 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.Word.PageControl
```

(例)　C#の場合

```
//eDocuToolDataBinder オブジェクトの作成（ページバージョン）
eDocuToolDataBinder.Word.PageControl eDocBinder = new eDocuToolDataBinder.Word.PageControl();
```

⑤eDocuTool.net の処理の流れ

ページバージョン

①ページ読み込み	startPage("ファイル名");
②ページの追加	setAddPage
③レポート内の項目名に値を設定	setTextData
④ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ページをクローズ	endPage("ファイル名");

1.startPage

Word 文書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 ページの追加	setAddPage
③ レポート内の項目名に値を設定	setTextData
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ ページをクローズ	endPage("ファイル名");

書式

edoc.startPage（"出力文書定義ファイル名"）

例　ファイル名が"ＡＢＣ"の場合

edoc.startPage ("ＡＢＣ")

ＶＢ例

```
edoc.startPage ("ＡＢＣ")
```

Ｃ＃例

```
edoc.startPage ("ＡＢＣ");
```

2.setAddPage

Word 文書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 ページの追加	setAddPage
③ レポート内の項目名に値を設定	setTextData
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ ページをクローズ	endPage("ファイル名");

書式

```
edoc.setAddPage ()
```

記述例

```
edoc.setAddPage ()
```

ＶＢ例

```
edoc.setAddPage ()
```

Ｃ＃例

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

※２回目以降の setAddPage は改ページを行う

3．setTextData

Word 文書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 ページの追加	setAddPage
③ レポート内の項目名に値を設定	**setTextData**
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ 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")

Word文書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。
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");
```

4.setTextDataAll

Word 文書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 ページの追加	setAddPage
③ レポート内の同項目名全てに値を設定	setTextDataAll
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ 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")

Word文書で作成した文字列の中に定義名標を"#項目１#"と定義した全ての名標に"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");
```

5 . setShapeTextData

図形の定義フィールドにテキストを転送

処理の流れ

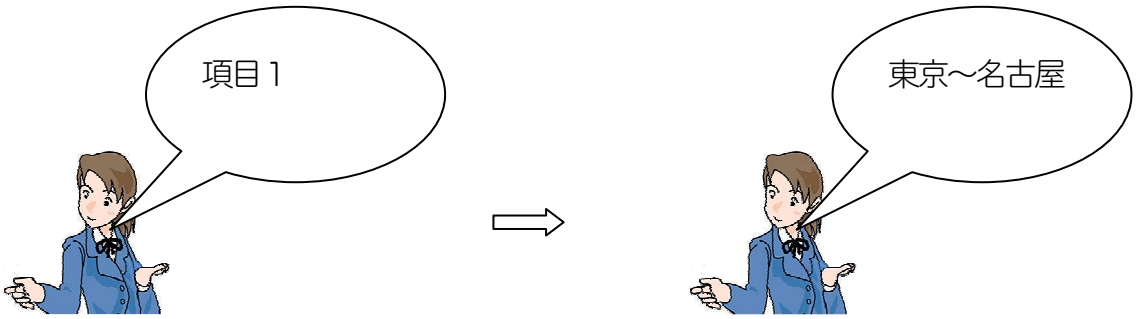
① ページ読み込み	startPage("ファイル名");
② ページの追加	setAddPage
③ レポート内の項目名に値を設定	**setShapeTextData**
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ ページをクローズ	endPage("ファイル名");

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 ")



図形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。
本ＤＬＬはストリング処理を行う機能である。

ＶＢ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 ") ;

6．setSmartArtTextData

SmartArt の定義フィールドにテキストを転送

処理の流れ

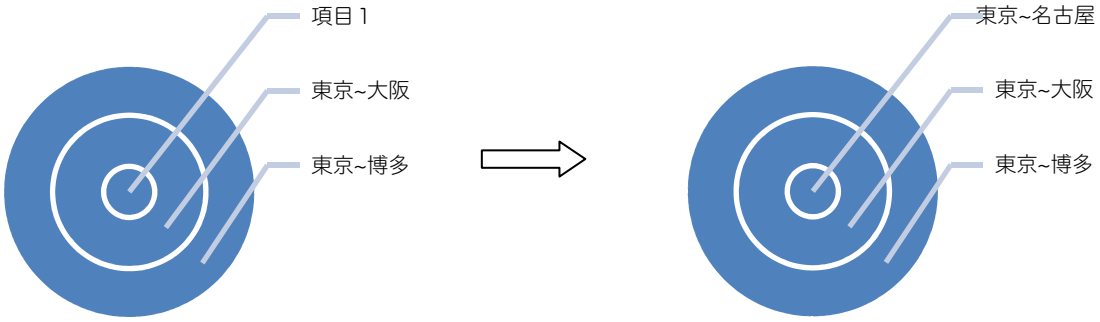
① ページ読み込み	startPage("ファイル名");
② ページの追加	setAddPage
③ レポート内の項目名に値を設定	**setSmartArtTextData**
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ ページをクローズ	endPage("ファイル名");

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")



SmartArt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。
本ＤＬＬはストリング処理を行う機能である。

ＶＢ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");
```

7 .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

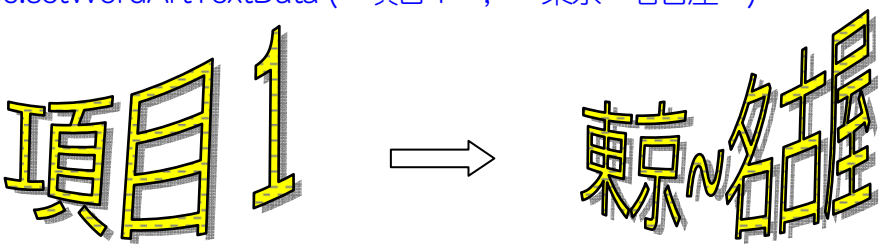
① ページ読み込み	startPage("ファイル名");
② ページの追加	setAddPage
③ レポート内の項目名に値を設定	**setWordArtTextData**
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ ページをクローズ	endPage("ファイル名");

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.setWordArtTextData (" 項目１ " ,　" 東京～名古屋 ")



ワードアート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。
本ＤＬＬはストリング処理を行う機能である。

ＶＢ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 ") ;

8.textDataClear

定義したＷｏｒｄ文書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、文書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 ページの追加	setAddPage
③ レポート内の項目名に値を設定	setTextData
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ ページをクローズ	endPage("ファイル名");

書式

edoc.textDataClear(" #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.textDataClear (" #ListArea# ")

この場合＂ＬｉｓｔＡｒｅａ＂とＷｏｒｄ文書で作成した全ての同名定義欄がクリアされる。

ＶＢ例

```
edoc.textDataClear ( " #出力定義項目# " )
```

Ｃ＃例

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

9.setPicData

Word 文書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 ページの追加	setAddPage
③ レポート内の同項目名全てに値を設定	setPicDatal
④ ページ内の項目名全てクリア	delPicData
⑤ 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

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・代替テキストで確認。

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") ;

10.setPicDataAll

Word 文書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 ページの追加	setAddPage
③ レポート内の同項目名全てに値を設定	setPicDataAll
④ ページ内の項目名全てクリア	delPicData
⑤ 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・代替テキストで確認。

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") ;

11.delPicData

Word 文書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 ページの追加	setAddPage
③ レポート内の項目名に値を設定	setPicData
④ ページ内の項目名全てクリア	**delPicData**
⑤ 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・代替テキストで確認。
この場合 " ＬｉｓｔＡｒｅａ " とWord文書で作成した全ての同名定義がクリアされる。

ＶＢ例

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

Ｃ＃例

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

12.endPage

Word 文書レポート出力の終了処理

処理の流れ

① ページ読み込み	startPage("ファイル名");
② ページの追加	setAddPage
③ レポート内の項目名に値を設定	setTextData
④ ページ内の項目名全てクリア	textDataClear("対象項目名");
⑤ ページをクローズ	**endPage("ファイル名");**

書式

edoc.endPage(出力文書定義ファイル名)

例　出力文書に全ての出力データが処理完となった時点で本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＂ ) ;
```

13.docxToPdf

DOCXファイルからPDFファイルへ変換

書式

```
moc.docxToPdf(string inputFileName, string outputPdfFilename,
              int outputPdfStartPage, int outputPdfEndPage,
              out string result, out string resultMessage)
```

inputFileName：DOCX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

outputPdfStartPage：出力開始ページ

outputPdfEndPage：出力終了ページ

result：結果　（「1」：成功　　「9」：失敗　）

resultMessage：resultが「9」の場合のメッセージ

※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

ＶＢ例

```
moc.docxToPdf( " DOCXファイル名 " ,  " PDFファイル名 " , 0, 0, out result, out resultmessage)
```

Ｃ＃例

```
moc.docxToPdf( " DOCXファイル名 " ,  " PDFファイル名 " , 0, 0, out result, out resultmessage);
```

14.docxToXps

DOCXファイルからXPSファイルへ変換

書式

```
moc.docxToXps(string inputFileName, string outputXpsFilename,
              int outputXpsStartPage, int outputXpsEndPage,
              out string result, out string resultMessage)
```

inputFileName：DOCX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

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

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.docxToXps( " DOCXファイル名 " ,  " XPSファイル名 " , 0, 0, out result, out resultmessage)
```

Ｃ＃例

```
moc.docxToXps( " DOCX ファイル名 " ,  " XPS ファイル名 " , 0, 0, out result, out resultmessage);
```

15.docxToDoc

DOCXファイルからDOCファイル（旧Word形式）へ変換

書式

```
moc.docxToXls(string inputFileName, string outputDocFilename,
              out string result, out string resultMessage)
```

inputFileName：DOCXファイルの絶対パス

outputDocFilename：DOC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

ＶＢ例

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

Ｃ＃例

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

16.docxToPrinter

DOCXファイルからプリンタ出力

書式

```
moc.docxToPrinter(string inputFileName, string outputPrintername,
                  int outputPrinterStartPage, int outputPrinterEndPage,
                  int outputPrinterCopies, out string result,
                  out string resultMessage)
```

inputFileName：DOCX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

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

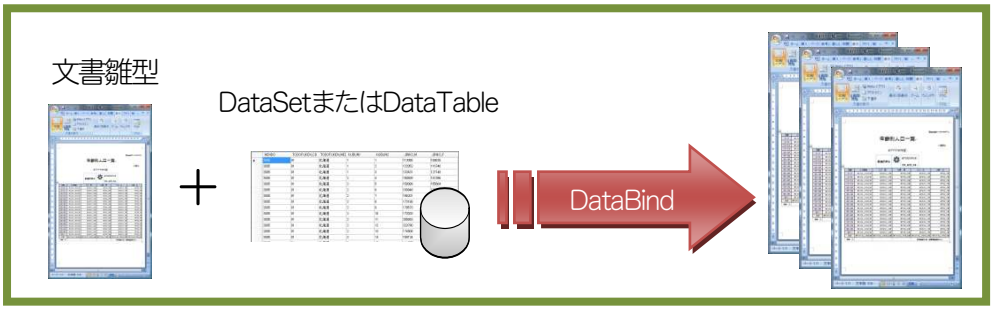
Ｃ＃例

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

17.eDocuToolDataBinder.Word.PageControl
（データバインド　ページバージョン）

概要

文書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文書作成を完結する。ステップ数の大幅な削減となる。



事前設定

文書雛型の設計時に、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
sourceDocxFileName	○	文書雛型ファイル名
outputDocxFileName	○	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）
DataSource	○	データソース（DataSetまたはDataTable）
TextDataBinding		Text設定でデータを帳票にセットするか（デフォルト：true）
ImageDataBinding		Image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）
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）
バインド時に設定する項目名	**必須**	**解説**
DataBind()	○	文書雛型とDataSourceを結合
バインド後に取得できる項目名	**必須**	**解説**
SuccessCount		データを文書にセット成功した件数
FailureCount		データを文書にセット失敗した件数
TextDataBindSuccessCount		Text設定でデータを文書にセット成功した件数
TextDataBindFailureCount		Text設定でデータを文書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.Word.PageControl
3. eDocBinder.sourceDocxFileName = @"c:\inetpub\baseDocu\年齢性別一覧_word2.docx"
4. eDocBinder.outputDocxFileName = @"c:\inetpub\wwwroot\outxlsx\年齢性別一覧_word2_" +
                                   DateTime.Now.ToString("yyyyMMddHHmmssfffffff") + ".docx"
5. eDocBinder.DataSource = db_TP5Y
6. eDocBinder.DataBind()
```

記述例（C#）

```
1. DataTable db_TP5Y = erdb.SqlDbGet(sqlStr, sqlc);
2. eDocuToolDataBinder.Word.PageControl eDocBinder = new eDocuToolDataBinder.Word.PageControl ();
3. eDocBinder.sourceDocxFileName = @"c:\inetpub\basexlsx\年齢性別一覧_word2.docx";
4. eDocBinder.outputDocxFileName = @"c:\inetpub\wwwroot\outxlsx\年齢性別一覧_word2_" +
                                   DateTime.Now.ToString("yyyyMMddHHmmssfffffff") + ".docx";
5. eDocBinder.DataSource = db_TP5Y;
6. 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. 文書雛型ファイル名の設定
4. 出力ファイル名の設定
5. データソースの設定
6. データバインド

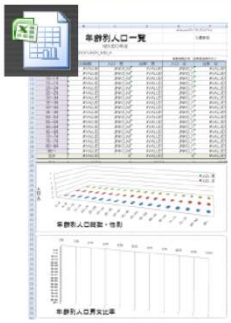
本ツールの利用方法

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)にするかを決める

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 | DataBind 記述

最小 6 命令の帳票プログラミング

```
//事業作成開始
edoc.startPage(xlsxFileName);
edoc.setNextSheet(xlsxSheetName);
edoc.setAddPage();

//CLIENTLOG_DBから展開
edoc.setPageTextData("YYYY", DDL_YM.SelectedItem.ToString());
edoc.setPageFieldData("TODOFUKEN_MEI_K", DDL_TODOFUKEN.SelectedItem.ToString());

//Excel内のラベルへデータ埋め込み
for (int i = 0; i < db_TP5Y.Rows.Count; i++)
{
    edoc.setPageFieldData("man", Convert.ToDecimal(db_TP5Y.Rows[i]["JINKO_M"]));
    edoc.setPageFieldData("woman", Convert.ToDecimal(db_TP5Y.Rows[i]["JINKO_F"]));
}

//帳票出力
edoc.saveSheet();

//ページ終了
edoc.endPage(xlsxFileName + "_" + now);
```

DataBind()命令のみの帳票プログラミング

（帳票と DataBese の結合を行います）

```
//バインド
eDocuTool.DataBinder.Excel.PageControl eDocBinder = new eDocuTool.DataBinder.Excel.PageControl();
eDocBinder.sourceXlsxFileName = "年齢性別一覧2";//帳票ファイル名
eDocBinder.sourceXlsxSheetName = "年齢別統計";//シート名
eDocBinder.outputXlsxFileName = "年齢性別一覧_" + now;//出力XLSXファイル名
//eDocBinder.ChangingPageRowCount = 15;
eDocBinder.ChangingPageConditionString = "NENDO,TODOFUKEN_CD";//改ページオプション
eDocBinder.OmissionBindingColumns = true;//処理効率化オプション
eDocBinder.DataSource = db_TP5Y;//レポート用DT
eDocBinder.DataBind();//データバインド
```

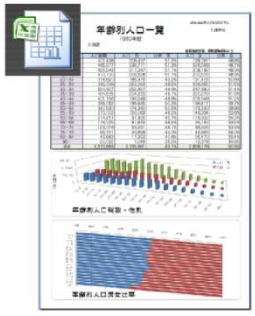
※サンプルプログラムは「Excel PageControl」を利用（帳票出力部分のみ抜粋）

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



Memo

Memo

Memo

eDocuTool.net©

Word 版マニュアル

株式会社R&Dソフトウエア
〒454-0012　名古屋市中川区尾頭橋四丁目１３番７号 nabi金山４０１
℡ 052-331-3871　URL http://www.rdsoftware.net