

GKitTaglib

# ExcelGenerator

( エクセルジェネレータ )

## マニュアル

20081022 版

# 目次

ExcelGenerator 概要.....	4
製品構成 .....	4
インストール配布物 .....	4
動作環境 .....	5
セットアップ .....	5
インストール方法 (Windows) .....	5
インストール方法 (Solaris・Linux) .....	7
Jakarta POI について .....	7
設定について .....	7
使い方 .....	8
用意するエクセルファイルについて .....	8
● 雛型ヘッダエクセルファイル .....	8
● 雛型レコードエクセルファイル .....	8
● 雛型フッタエクセルファイル .....	9
サポートエクセルファイル機能一覧 .....	9
EGRange の取得について .....	13
データ設定について .....	14
● boolean データの設定 .....	14
● Calendar データの設定 .....	14
● Date データの設定 .....	15
● 数値データの設定 .....	15
● 文字列データの設定 .....	15
データ取得について .....	16
● セルタイプの取得 .....	16
● Boolean データの取得 .....	17
● Calendar データの取得 .....	17
● Date データの取得 .....	17
● Double データの取得 .....	17
● String データの取得 .....	17
● 式の取得 .....	18
● 最終行番号の取得 .....	18
● 最終列番号の取得 .....	18
印刷設定について .....	19
● 印刷枚数の設定 .....	19

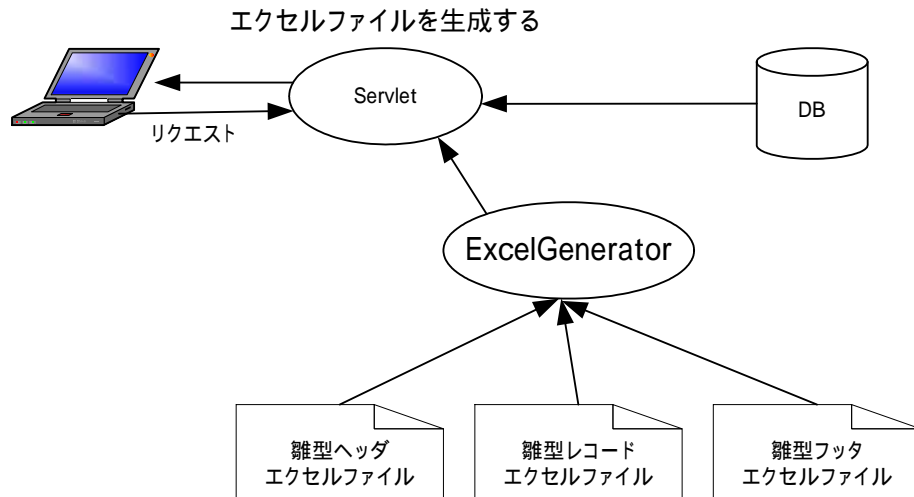
● 簡易印刷の設定 .....	19
● 印刷用紙サイズの設定 .....	19
● 印刷フィット高さの設定 .....	20
● 印刷フィット幅の設定 .....	20
● 印刷フッタ余白の設定 .....	21
● 印刷ヘッダ余白の設定 .....	21
● 印刷向きの設定 .....	22
● ページ方向の設定 .....	22
● 上余白の設定 .....	22
● 下余白の設定 .....	23
● 左余白の設定 .....	23
● 右余白の設定 .....	24
● 白黒印刷の設定 .....	24
● 印刷品質の設定 .....	24
● 印刷ヘッダ・フッタの設定 .....	25
● 改ページの設定 .....	26
● 印刷タイトルの設定 .....	26
背景色設定について .....	27
フォント設定について .....	27
● 色設定 .....	27
● フォント名設定 .....	27
● フォントサイズ設定 .....	27
● ボールド設定 .....	27
● イタリック設定 .....	28
● アンダーライン設定 .....	28
式の設定について .....	29
● 式の設定 .....	29
● 式の自動補正機能について .....	29
出力しないシートについて .....	30
シートのレイアウト設定について .....	30
● ウィンドウ枠の固定 .....	30
● 行高さの設定 .....	30
チュートリアル .....	31
仕様 .....	31

雛型エクセルの準備 .....	31	印刷ヘッダ・フッタを付けたい .....	36
プログラム .....	32	開発中にエクセルで雛型ファイルが開かなくなるときがある .....	36
実行 .....	34	Javadoc .....	36
HOW TO .....	35	ライセンス .....	37
雛型シートの内容が表示されない .....	35	Jakarta POI ライセンス .....	37
レコード部またはフッタ部が必要ない .....	35	ExcelGenerator ライセンス .....	38
出力されるシート数をプログラムで増やす .....	35	サポートについて .....	39
印刷範囲を1枚に収めるには? .....	36	無償サポート .....	39
サブレットにブラウザでアクセスしてもエクセルが開かない .....	36	有償サポート .....	39
雛型エクセルファイルに設定した内容が出力エクセルファイルに反映されない .....	36	制限事項 .....	40

## ExcelGenerator 概要

ExcelGenerator ( エクセルジェネレータ ) は Jakarta POI を使用し Java プログラムでエクセルファイルを生成するためのライブラリです。雛型エクセルを用意し、エクセルでの帳票やデータの一覧表などの作成に向いています。Web アプリケーションで使用されることを想定しています。

雛型ファイルはヘッダ部、レコード部、フッタ部となるエクセルファイルを用意することで帳票や一覧表などの作成を行うことができます。



## 製品構成

インストールされる製品構成および動作環境を説明します。

### インストール配布物

以下のフォルダ構成でモジュールの配布が行われます。

#### [インストールフォルダ]

[doc].....ドキュメントが格納されています。

[javadoc] .....javadoc が格納されています。

#### [WEB-INF]

[lib] .....ExcelGenerator で使用するライブラリが格納されています。

[src] .....

[sample] .....サンプルソースが格納されています。

[tlds] .....タグライブラリを使用する際の TLD が格納されています。

[tmp] .....

[sample] .....サンプルソースで使用する雛型エクセルファイルがあります。

## 動作環境

下記の動作環境をサポートしています。

O S : Windows Server 2003

RedHat Linux Enterprise Edition version 3

Solaris 9 ( SPARC 版 )

A S : Fujitsu Interstage Aplication Server V6.0/V7.0/V8.0/V9.0

BEA WebLogic Server 8.1J/9.2J/10.0J

IBM WebSphere Aplication Server V5.0/V6.1

Jakarta Tomcat 4.1/5.5/6.0

Servlet API2.2 JSP1.1 に準拠するコンテナ

JDK : 1.3.1\_06 以上

POI : 必ず製品付属の POI をご利用ください

<http://jakarta.apache.org/poi/index.html>

C P U : 500MHz 以上

メモリ : 512 MB 以上

## セットアップ

ここでは各 O S でのセットアップ手順について説明いたします。

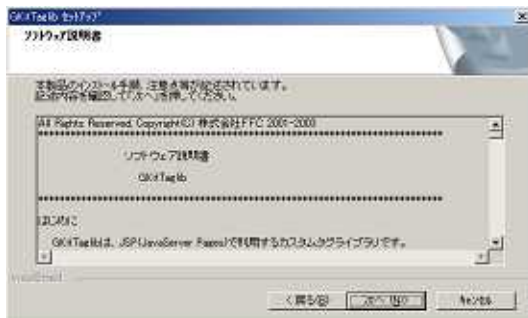
### インストール方法 ( Windows )

インストールする端末に C D をいれるとインストーラが自動的に起動します。

自動的に起動しない場合は C D 内の INSTALL.exe を実行してください。



ソフトウェア説明書が表示されるので一読してください。



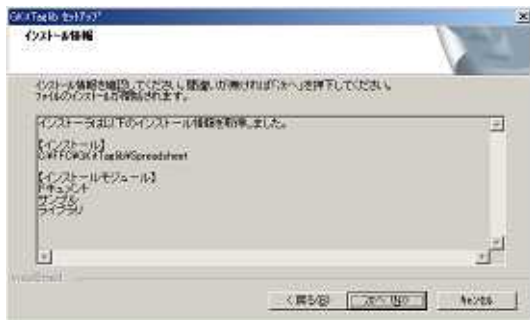
インストール先を選択してください。



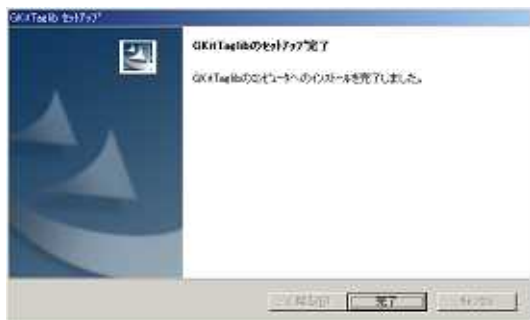
インストールするコンポーネントを選択してください。



インストール情報が表示されますので確認してください。次へを押下するとファイル転送が始まります。



完了画面が表示されるとインストールは終了です。



インストールフォルダを開きモジュールが配布されているか確認してください。

## インストール方法 (Solaris・Linux)

Solaris・Linux でインストールを行うために CD の tar フォルダに TAR ファイルを用意してあります。tar コマンドでファイルを展開してください。

インストールされるものは Windows インストーラでインストールされるものと同一です。

## Jakarta POI について

Jakarta POI は Java で Excel や Word 等のファイルを扱うためのライブラリです。詳しくは下記 URL をご覧ください。

<http://jakarta.apache.org/poi/index.html>

ExcelGenerator を動作させるためには Jakarta POI が必要になります。ExcelGenerator をご使用になる端末の CLASSPATH に POI ライブラリのパスを設定してください。

## 設定について

ExcelGenerator を使用するためには Jakarta POI および ExcelGenerator のライブラリの CLASSPATH を設定する必要があります。ここでは Jakarta POI の入手および設定、ExcelGenerator の設定について説明します。

### [POI ライブラリの入手]

下記 URL より POI のモジュールを入手してください。

<http://jakarta.apache.org/poi/index.html>

### [POI の設定]

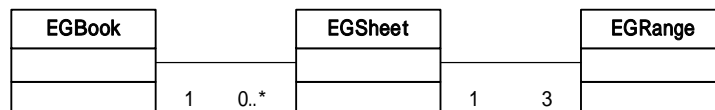
入手したモジュールを解凍し POI の JAR ファイルを取り出し、任意の場所に配置してください。環境変数 CLASSPATH に POI の JAR のパスを設定します。

### [ExcelGenerator の設定]

インストールフォルダにある” excelgenerator-1.0.jar”のパスを環境変数 CLASSPATH に設定します。

## 使い方

ここでは ExcelGenerator の詳細な使い方を説明いたします。ExcelGenerator は EGBok、EGSheet、EGRange のクラスを使用し、データ設定していきます。クラス図は下記のようになっています。EGBok (出力エクセルファイルを表現) の中に EGSheet (各シートを表現) があり、EGSheet の中にヘッダ、レコード、フッタを表現する EGRange が存在します。



ExcelGenerator で使用できる API は製品のインストールフォルダにある Javadoc を参照してください。

## 用意するエクセルファイルについて

ExcelGenerator を使用するには Jakarta POI の他に、雛型となるエクセルファイルを用意していただく必要があります。雛型エクセルファイルは Excel97 形式で作成する必要があります。雛型ヘッダエクセルファイル、雛型レコードエクセルファイル、雛型フッタエクセルファイルで出力されるエクセルファイルを定義します。

氏名	従業員番号	職種	電話番号
田中 三太郎	0007654	054-3210 東京駅前ビル5丁目	03-0123-4567
山田 次郎	0004321	067-0123 埼玉県さいたま市西2丁目	03-0871-6245
佐々木 一郎	0019876	021-0088 千葉県船橋市北2丁目	03-4354-4568
鈴木 五三郎	0023456	087-0024 神奈川県横浜市西区4丁目	03-5555-4545
高橋 四五郎	0456789	125-0048 群馬県高崎市西4丁目	03-4545-4545
伊藤 三太郎	0881023	345-0041 茨城県水戸市5丁目4番地	03-4545-4565
宇野 一郎	0043210	320-0123 栃木県宇都宮市1丁目	03-4578-4579
石川 川治郎	0667890	021-0079 東京都港区南1丁目	03-1245-1232
渡辺 三太郎	1145678	031-0047 東京都港区北1丁目	03-1255-1235
横山 三太郎	0012345	115-0076 千葉県千葉市中央区1丁目	03-8865-8865
森田 三太郎	0076543	031-0074 埼玉県さいたま市東区2丁目	03-4578-4589

出力エクセルファイル

部内アドレス帳			
氏名	従業員番号	職種	電話番号
田中 三太郎	0007654	054-3210 東京駅前ビル5丁目	03-0123-4567

部内アドレス帳			
氏名	従業員番号	職種	電話番号
田中 三太郎	0007654	054-3210 東京駅前ビル5丁目	03-0123-4567

部内アドレス帳			
氏名	従業員番号	職種	電話番号
田中 三太郎	0007654	054-3210 東京駅前ビル5丁目	03-0123-4567

雛型ヘッダエクセルファイル

雛型レコードエクセルファイル

雛型フッタエクセルファイル

### ● 雛型ヘッダエクセルファイル

雛型ヘッダエクセルファイルは出力されるエクセルファイルの元になるものです。ヘッダファイルのセル幅、印刷設定、生成されるシート数が出力ファイルに引継がれます。各シートに対しヘッダとなる部分の設定をします。

### ● 雛型レコードエクセルファイル

雛型レコードエクセルファイルは繰り返される部分の1レコード分を表したものになります。1レコード分の設定を行います。ヘッダファイルのシート番号(シートの左から0、1、2……)と同じレコードファイルのシート番号のシートが組みになります



## ● 雛型フッタエクセルファイル

雛型フッタエクセルファイルは出力エクセルファイルのフッタ部分を表します。フッタ部分の設定を行います。ヘッダファイルのシート番号と同じフッタファイルのシート番号のシートが組みになります。

### サポートエクセルファイル機能一覧

雛形となるエクセルファイルの機能一覧です。雛型が×となっている項目は非推奨または出力ファイルに反映付加の項目です。

A P I が となっている項目は ExcelGenerator の A P I が用意してある項目です。

下記の表はエクセルのメニューから機能を抜き出しています。

エクセル機能一覧				雛型	A P I
セルの書式設定					
	表示形式				
	配置				
		横位置			
			標準		
			左詰め（インデント）		
			中央揃え		
			右詰め		
			繰り返し		
			両端揃え		
			選択範囲内で中央		
		縦位置			
			上詰め		
			中央揃え		
			下詰め		
			両端揃え		
			均等割り付け		
		前後にスペースを入れる		×	
		折り返して全体を表示する			
		縮小して全体を表示する			
		セルを結合する			
		インデント			
		縦書き			
		角度			
	フォント				
		フォント名			
		スタイル			

	サイズ		
	下線		
	色		
	標準フォント		
	取り消し線		
	上付き		
	下付き		
	罫線		
	上罫線		
	下罫線		
	右罫線		
	左罫線		
	縦罫線		
	横罫線		
	右下罫線	×	
	左下罫線	×	
	色		
	スタイル		
	パターン		
	前景色		
	背景色		
	パターン		
	保護		
	ロック	×	1
	表示しない	×	1
ページ設定			
	ページ		
	印刷の向き		
	拡大縮小%		
	縦×横		
	用紙サイズ		
	印刷品質		
	先頭ページ番号		
	余白		
	上		
	下		
	右		
	左		

		ヘッダ		
		フッタ		
		水平		
		垂直		
	ヘッダ / フッタ			
		ヘッダ	×	
		フッタ	×	
	シート			
		印刷範囲	×	
		行のタイトル	×	
		列のタイトル	×	
		枠線		
		行列番号	×	
		白黒印刷		
		簡易印刷		
		コメント	×	
		ページ方向		
印刷				
	プリンタ		×	
	すべて		×	
	ページ指定		×	
	印刷対象		×	
	部数		×	
	部単位で印刷		×	
プロパティ				
	ファイルの概要			
		タイトル	×	
		サブタイトル	×	
		作成者	×	
		管理者	×	
		会社名	×	
		分類	×	
		キーワード	×	
		コメント	×	
		ハイパーリンクの基点	×	
		プレビューの図を保存する	×	
表示				
	改ページプレビュー		×	

	コメント		×	
	全画面表示		×	
	ズーム		×	
挿入				
	グラフ		×	
	改ページ		×	
	関数			
	名前			
		定義	×	
	図		×	
	オブジェクト		×	
	ハイパーリンク			
書式				
	行			
		高さ		
		表示しない	×	
	列			
		幅		
		表示しない		
		標準幅	×	
	シート			
		名前		
		背景	×	
		表示しない	×	
	条件付書式		×	
	スタイル			
	ふりがな		×	
ツール				
	ブックの共有		×	
	保護		×	
	ゴールシーク		×	
	シナリオ		×	
	マクロ		×	1
	アドイン		×	
データ				
	入力規則		×	1
	テーブル		×	
	グループ化		×	

	アウトライン設定	×	
ウィンドウ			
	分割	×	
	ウィンドウ枠の固定	×	
コントロール ツールボックス			
	コントロール	×	
フォーム		×	
ワードアート		×	1
図		×	1
図形		×	1

1 データの読み込みは可能。

#### EGRange の取得について

データを取得・設定するためには EGRange オブジェクトを取得する必要があります。EGRange オブジェクトは各シートそれぞれにヘッダ、レコード、フッタの EGRange があります。

##### ・EGSheet#getHeaderRange()

EGBook のコンストラクタの第 1 引数で渡される雛型ヘッダエクセルファイルに対する EGRange を取得します。対象となるシートは EGSheet が対象としているシートになります。

```
//EGBook 生成
EGBook book = new EGBook(ヘッダエクセル, レコードエクセル, フッタエクセル);
//対象となるシートの EGSheet の取得
EGSheet sheet = book.getSheetAt(2);
//ヘッダエクセルの EGRange を取得
EGRange range = sheet.getHeaderRange();
```

##### ・EGSheet#getRecordRange( index )

EGBook のコンストラクタの第 2 引数で渡される雛型レコードエクセルファイルに対する EGRange を取得します。対象となるシートは EGSheet が対象としているシートになります。各シートのレコード毎に EGRange があります。

```
//EGBook 生成
EGBook book = new EGBook(ヘッダエクセル, レコードエクセル, フッタエクセル);
//対象となるシートの EGSheet の取得
EGSheet sheet = book.getSheetAt(2);
//レコードエクセルの EGRange を取得
EGRange range = sheet.getRecordRange(3);
```

- EGSheet#getFooterRange()

EGBook のコンストラクタの第 3 引数で渡される雛型フッタエクセルファイルに対する EGRange を取得します。対象となるシートは EGSheet が対象としているシートになります。

```
//EGBook 生成
EGBook book = new EGBook(ヘッダエクセル, レコードエクセル, フッタエクセル);
//対象となるシートの EGSheet の取得
EGSheet sheet = book.getSheetAt(2);
//フッタエクセルの EGRange を取得
EGRange range = sheet.getFooterRange();
```

### データ設定について

データ設定は EGRange クラスを通して行います。EGRange#setData(row, col, value)メソッドでデータ設定することができます。row、col は雛型エクセルファイルでの行列数を指定します。エクセルファイルの C2 のセルを指定するときは setData(1, 2, value)と設定します。

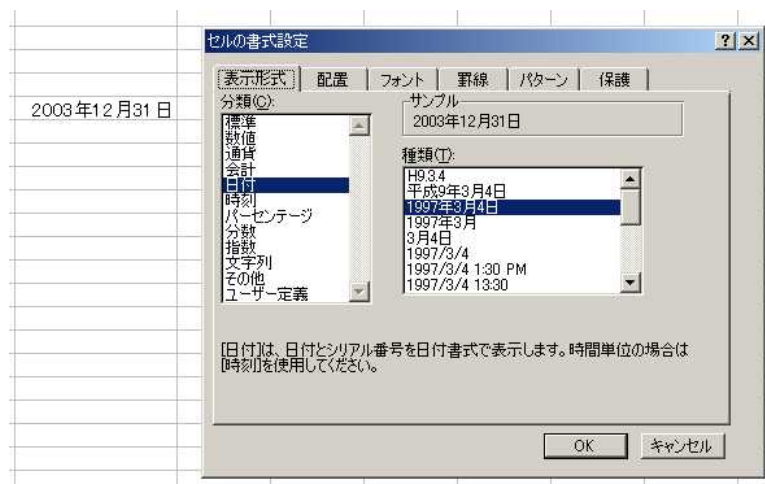
- boolean データの設定

EGRange#setData(row, col, **boolean**)とすることで true、false の設定ができます。

	TRUE	FALSE
--	------	-------

- Calendar データの設定

EGRange#setData(row, col, **Calendar**)とすることで日付型のセルに日付の入力ができます。



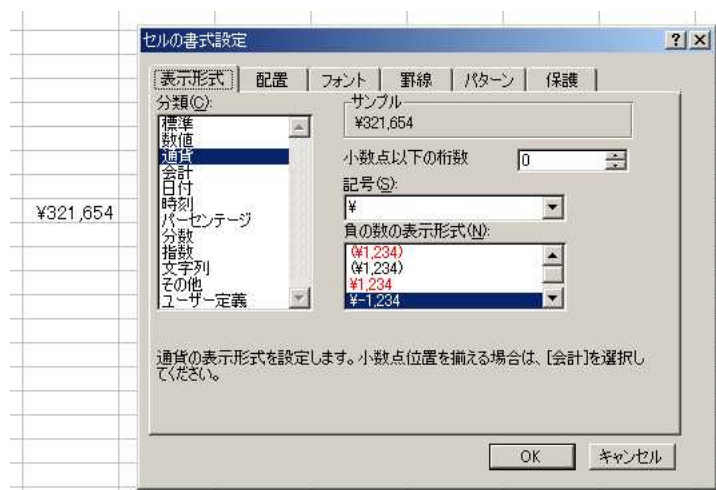
### ● Date データの設定

EGRANGE#setData(row, col, **Date**)とすることで時刻型のセルに時刻の入力ができます。



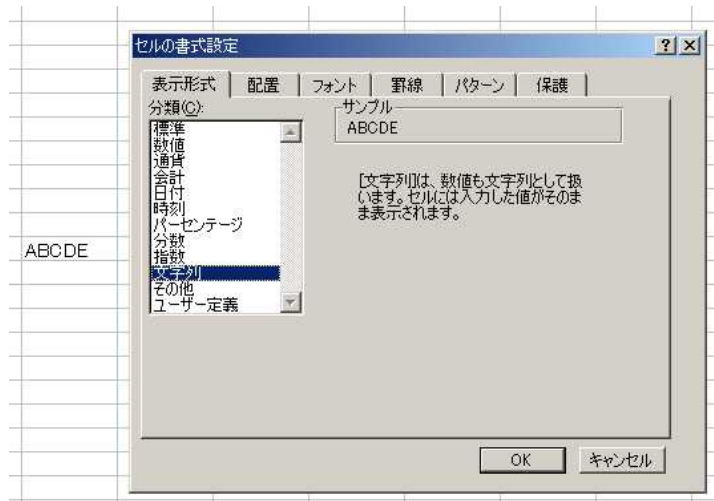
### ● 数値データの設定

EGRANGE#setData(row, col, **double**)とすることで数値・通貨型のセルに数値の入力ができます。



### ● 文字列データの設定

EGRANGE#setData(row, col, **String**)で文字列のセルに文字列の入力ができます。



## データ取得について

データの取得は EGRRange クラスを通して行います。get ~ (row, col) メソッドでデータを取得することができます。

### ●セルタイプの取得

EGRRange#getType(row, col) でセルタイプを Short で取得することができます。返されるセルタイプによりデータ取得メソッドを使い分けます。

セルが存在しない場合は null が返ります。

### [セルタイプ種別]

- EGRRange . CELL\_TYPE\_BLANK  
データは設定されていません。
- EGRRange . CELL\_TYPE\_STRING  
String が設定されています。
- EGRRange . CELL\_TYPE\_BOOLEAN  
boolean が設定されています。
- EGRRange . CELL\_TYPE\_NUMERIC  
double が設定されています。
- EGRRange . CELL\_TYPE\_CALENDAR  
Calendar が設定されています。
- EGRRange . CELL\_TYPE\_DATE  
Date が設定されています。
- EGRRange . CELL\_TYPE\_FORMULA  
式が設定されています。

Short 型の戻り値を以下のようにしてセルタイプを判定してください。



```

Short type = ERange . getType(2, 3);
if( type.shortValue() == ERange.CELL_TYPE_STRING ){
    //String が設定されています。
}

```

- Boolean データの取得

ERange# getDataBoolean(int row, int col)で Boolean 型のデータを取得できます。

[使用可能セルタイプ]

CELL\_TYPE\_BOOLEAN

- Calendar データの取得

ERange# getDataCalendar(int row, int col)で Calendar 型のデータを取得できます。

雛型エクセルファイルに日付型を指定した場合でも

ERange# setData(int row, int col, java.util.Calendar value)を呼び出していない場合はセルタイプが CELL\_TYPE\_NUMERIC となります。

[使用可能セルタイプ]

CELL\_TYPE\_NUMERIC

CELL\_TYPE\_CALENDAR

- Date データの取得

ERange# getDataDate(int row, int col)で Date 型のデータを取得できます。

雛型エクセルファイルに日付型を指定した場合でも

ERange# setData(int row, int col, java.util.Date value)を呼び出していない場合はセルタイプが CELL\_TYPE\_NUMERIC となります。

[使用可能セルタイプ]

CELL\_TYPE\_NUMERIC

CELL\_TYPE\_DATE

- Double データの取得

ERange# getDataNumeric(int row, int col)で Double 型のデータを取得できます。

[使用可能セルタイプ]

CELL\_TYPE\_NUMERIC

- String データの取得

ERange# getDataString(int row, int col)で String 型のデータを取得できます。

[使用可能セルタイプ]  
CELL\_TYPE\_STRING

- 式の取得

EGRange# getFormula(int row, int col)で式の値を String 型で取得できます。

[使用可能セルタイプ]  
CELL\_TYPE\_FORMULA

- 最終行番号の取得

EGRange# getLastRowNum()でセルの存在する最終行の番号を取得することができます。

データやスタイルなどが設定されているセルが存在するセルとなります。

シートが以下のようにになっている場合は 3 が取得されます。

	A	B	C
1	0	1	2
2	3	4	5
3		7	
4			



- 最終列番号の取得

EGRange# getLastCellNum(int rownum)で指定した行のセルの存在する最終列の番号を取得することができます。

データやスタイルなどが設定されているセルが存在するセルとなります。

シートが以下のようにになっている場合は

	A	B	C
1	0	1	
2	3	4	5
3		7	
4			

EGRange#getLastCellNum(0) では 2 が返ります。

EGRange#getLastCellNum(1) では 3 が返ります。

EGRange#getLastCellNum(2) では 2 が返ります。

EGRange#getLastCellNum(3) では 0 が返ります。

## 印刷設定について

印刷設定は雛型ヘッダエクセルファイルの設定を引継ぎます。プログラムで印刷設定を行う場合は EGSheet クラスのメソッドを使用します。

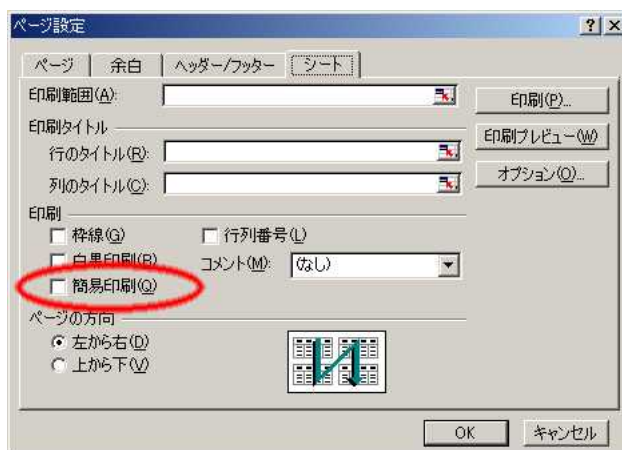
### ● 印刷枚数の設定

EGSheet #setCopies(short)で印刷枚数の設定を行います。



### ● 簡易印刷の設定

EGSheet #setDraft(boolean)で簡易印刷の設定を行います。

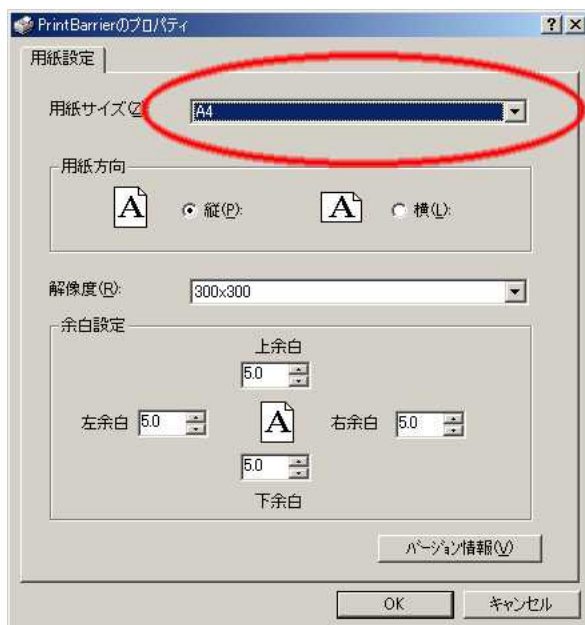


### ● 印刷用紙サイズの設定

EGSheet #setPaperSize(short)で印刷用紙サイズの設定を行います。

EGSheet の静的な値として用紙サイズ値が用意されています。用意されている用紙サイズは下記の通りです。

A1 サイズ、A2 サイズ、A3 サイズ、A4 サイズ、A5 サイズ、B4 サイズ、B5 サイズ、B6 サイズ、LEGAL サイズ、LETTER サイズ、はがきサイズです。



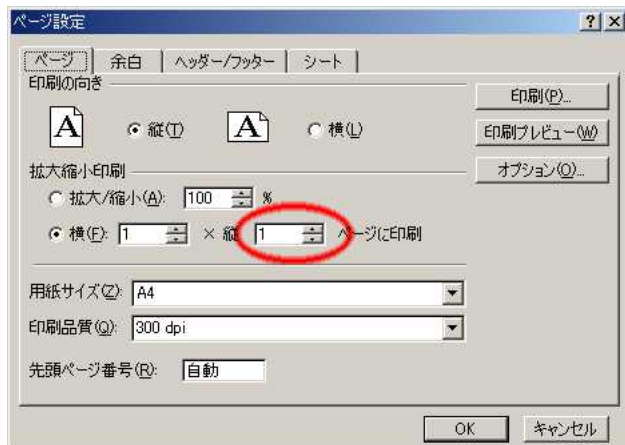
上記の画像は弊社から販売している PrintBarrier のプロパティ設定画面です。この画面はお客様の環境によって変わります。

PrintBarrier に関しては下記 URL をご覧ください。

<http://www.ffc.co.jp/products/printbarrier/index.html>

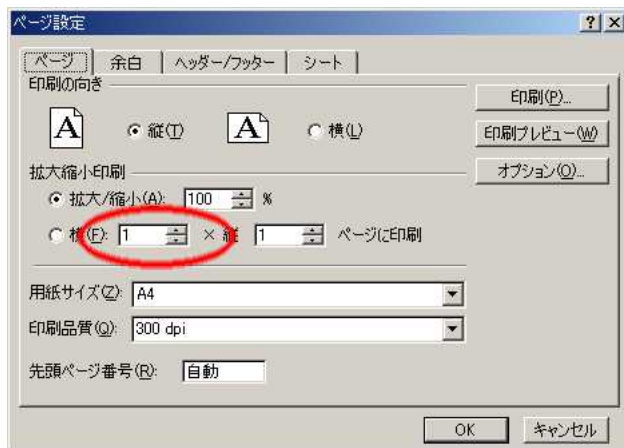
#### ● 印刷フィット高さの設定

EGSheet #setFitHeight(short)で印刷拡大縮小の縦の数を設定します。



#### ● 印刷フィット幅の設定

EGSheet #setFitWidth(short)で印刷拡大縮小の横の数を設定します。



### ● 印刷フッタ余白の設定

EGSheet#setHeaderMargin(double)で印刷余白フッタの設定を行います。引数はセンチ単位で設定してください。



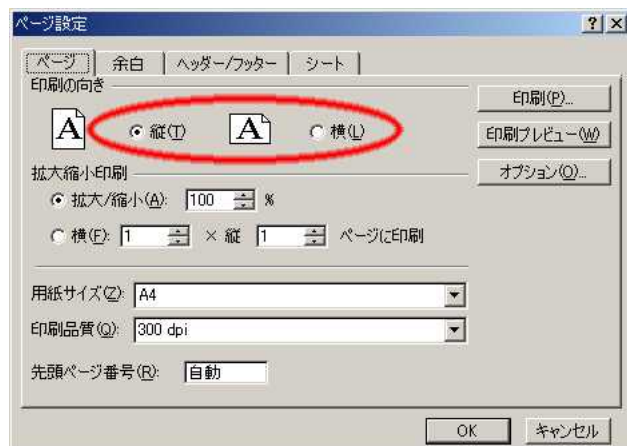
### ● 印刷ヘッダ余白の設定

EGSheet#setHeaderMargin(double)で印刷余白ヘッダの設定を行います。引数はセンチ単位で設定してください。



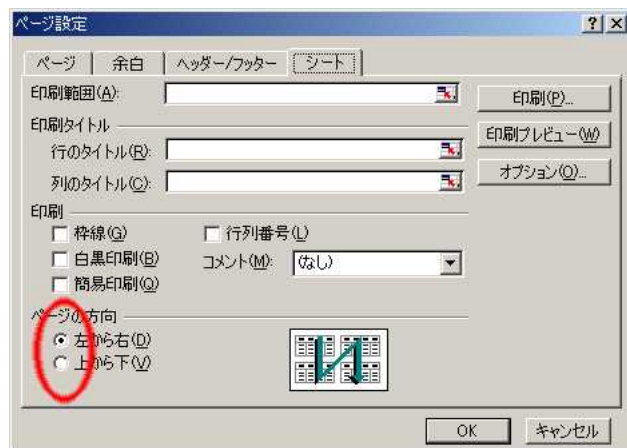
- 印刷向きの設定

EGSheet#setLandscape(boolean)で印刷向きを設定を行います。true の場合は横に設定されます。



- ページ方向の設定

EGSheet#setLeftToRight(boolean)でページ方向の設定を行います。true の場合は上から下に設定されます。



- 上余白の設定

EGSheet#setMargintop(double)で印刷余白上の設定を行います。引数はセンチ単位で設定してください。



### ● 下余白の設定

EGSheet#setMarginbottom(double)で印刷余白下の設定を行います。引数はセンチ単位で設定してください。



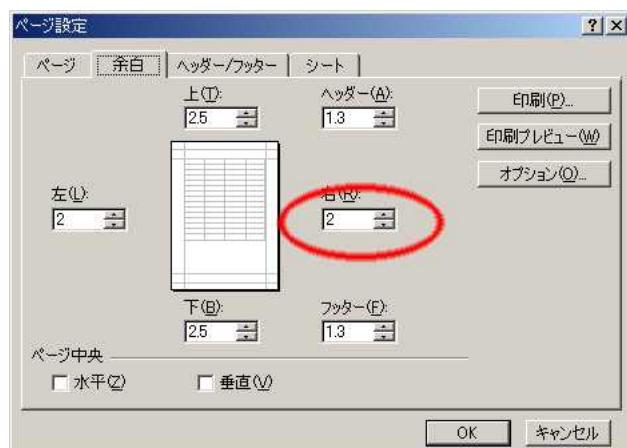
### ● 左余白の設定

EGSheet#setMarginleft(double)で印刷余白左の設定を行います。引数はセンチ単位で設定してください。



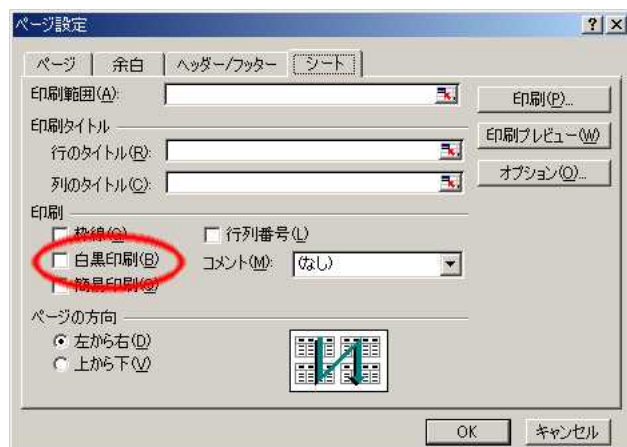
- 右余白の設定

EGSheet #setMarginRight(double)で印刷余白右の設定を行います。引数はセンチ単位で設定してください。



- 白黒印刷の設定

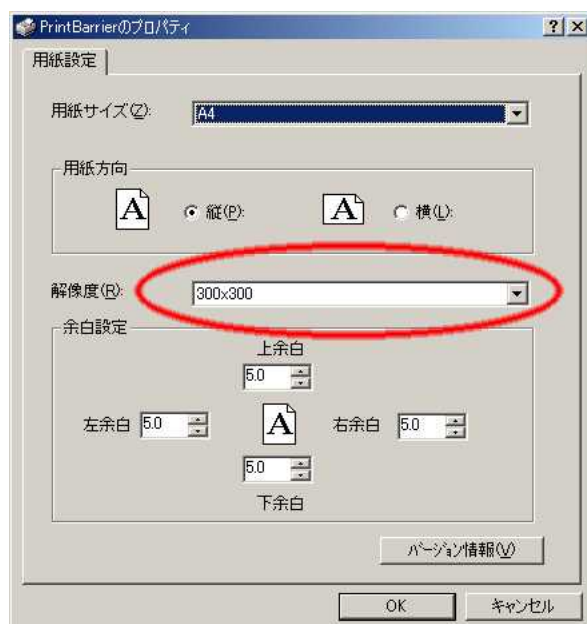
EGSheet #setNoColor(boolean)で白黒印刷の設定を行うことができます。



- 印刷品質の設定

EGSheet #setQuality(short)で印刷品質（解像度）の設定を行います。設定値は dpi 単位で設定してください。





上記の画像は弊社から販売している PrintBarrier のプロパティ設定画面です。この画面はお客様の環境によって変わります。

PrintBarrier に関しては下記 URL をご覧ください。

<http://www.ffc.co.jp/products/printbarrier/index.html>

#### ● 印刷ヘッダ・フッタの設定

印刷ヘッダ・フッタを設定するためには以下のメソッドを使用します。

EGSheet#setHeaderCenter(String).....ヘッダ中央  
 EGSheet#setHeaderRight(String).....ヘッダ右側  
 EGSheet#setHeaderLeft(String) .....ヘッダ左側  
 EGSheet#setFooterCenter(String).....フッタ中央  
 EGSheet#setFooterRight(String).....フッタ右側  
 EGSheet#setFooterLeft(String) .....フッタ左側

全角文字は文字化けを起こすため特定文字列以外で全角文字を使用することはできません。

#### ・ ページ番号などの設定

特定の文字列を含めることでページ番号などの設定を行うことができます。

説明	文字列 1	文字列 2
ページ番号を設定します	&[ページ番号]	&P
総ページ数を設定します	&[総ページ数]	&N
日付を設定します	&[日付]	&D
時刻を設定します	&[時刻]	&T
ファイルパスを設定します	&[パス]	&Z
ファイル名を設定します	&[ファイル名]	&F
シート名を設定します	&[シート名]	&A

・例

「1 / 3」のように「ページ番号 / 総ページ数」をフッタ中央に設定する場合

```
String footer = "&[ページ番号] / &[総ページ数]";
```

```
EGSheet . setFooterCenter( footer );
```

以下のように設定することも可能です。

```
String footer = "&P / &N";
```

```
EGSheet . setFooterCenter( footer );
```

## ● 改ページの設定

改ページを設定するためには EGSHEET# addPrintBreak(row, col)を使用します。

row : 行番号を設定します。

col : 列番号を設定します。

・例

以下のように改ページを設定する場合

	A	B	C	D	E	F	G
1							
2							
3							
4							

改ページを設定する

//C2 に改ページを挿入する

```
EGSheet . addPrintBreak(0, 1);
```

//G4 に改ページを挿入する

```
EGSheet . addPrintBreak(2, 5);
```

## ● 印刷タイトルの設定

印刷タイトルは以下のメソッドで設定を行うことができます。

・ EGSHEET#setTitleRow(row, length)

row : 行番号

length : タイトル行の長さ

・ EGSHEET#setTitleCol(col, length)

col : 列番号

length : タイトル列の長さ

・例

E-F 列、2-3 行を印刷タイトルに設定する場合

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								

//E-F 列を印刷タイトルに設定する ( 4 列目から 2 列分 )

```
EGSheet . setTitleCol(4, 2);
```

//2-3 行を印刷タイトルに設定する ( 2 行目から 2 列分 )

```
EGSheet . setTitleRow(1, 2);
```

### 背景色設定について

EGRange#setForegroundColor(row, col, color)を使用することによりセルの背景色の設定を行うことができます。設定できる色は EGRange に静的な変数として用意してあります。

### フォント設定について

フォント設定はセルのフォントの状態の設定を行います。セル全体に対し行います。セル内での異なるフォント設定に関してはサポートしておりません。また、雛形ファイルでのセル内の異なるフォンの設定はサポートしておりません。

#### ● 色設定

EGRange#setFontColor(row, col, color)を使用することにより文字色を設定することができます。設定できる色に関しては EGRange に静的な変数として用意してあります。

例) 4 行 5 列目のフォントを赤くする場合

```
EGRange.setFontColor( 4, 5, EGRange. COLOR_RED );
```

#### ● フォント名設定

EGRange#setFontName(row, col, name)を使用することによりフォントを設定することができます。設定するフォントの名前を第 3 引数に設定してください。

#### ● フォントサイズ設定

EGRange#setFontSize(row, col, size)を使用することによりフォントサイズを設定することができます。設定するサイズを第 3 引数に設定してください。

#### ● ボールド設定

EGRange#setFontBold(row, col, boolean)を使用することによりフォントのボールド設定を行うことができます。

- イタリック設定

EGRange#setFontItalic(row, col, boolean)を使用することによりフォントのイタリック設定を行うことができます。

- アンダーライン設定

EGRange#setFontUnderline(row, col, boolean)を使用することによりフォントのアンダーライン設定を行うことができます。

## 式の設定について

### ● 式の設定

EGRange# setFormula(row, col, value)を使用することでセルに式を設定することができます。

row (行) col (列) は雛型エクセルでのセルの行列番号を設定します。

value (式) は出力エクセルでの式を文字列で設定します。

### ● 式の自動補正機能について

EGBook#setAutoFormula(value)を使用することで式を自動的に補正することができます。  
レコードの増減により式の参照セルが変更する場合に対応することができるようになります。

value : true を設定することにより式の自動補正機能を有効にします。

#### 【自動補正有効時の動作】

自動補正有効時は雛型レコード、雛型フッタに設定してある式が補正されます。

有効ではない場合

##### ・雛型ヘッダ

=D1	=E1	=F1
=D2	=E2	=F2

##### ・雛型レコード

=D1	=E1	=F1
=D2	=E2	=F2

##### ・雛型フッタ

=D1	=E1	=F1
=D2	=E2	=F2

##### ・出力ファイル

=D1	=E1	=F1
=D2	=E2	=F2
=D1	=E1	=F1
=D2	=E2	=F2
=D1	=E1	=F1
=D2	=E2	=F2
=D1	=E1	=F1
=D2	=E2	=F2

有効な場合

##### ・雛型ヘッダ

=D1	=E1	=F1
=D2	=E2	=F2

##### ・雛型レコード

=D1	=E1	=F1
=D2	=E2	=F2

##### ・雛型フッタ

=D1	=E1	=F1
=D2	=E2	=F2

##### ・出力ファイル

=D1	=E1	=F1
=D2	=E2	=F2
=D3	=E3	=F3
=D4	=E4	=F4
=D5	=E5	=F5
=D6	=E6	=F6
=D7	=E7	=F7
=D8	=E8	=F8

自動補正機能を有効にすることでヘッダの行数やレコードの数に合わせて式が変化します。

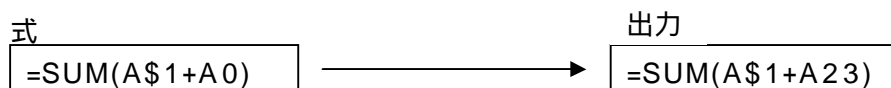
式と判断されるのは以下の場合です。

- ・ 式を設定した場合
- ・ セルの型を文字列にして=(イコール)で始まる文字を記述した場合

#### 式補正例

式	補正例
=A 1	=A 4
=A 1+2	=A 4+2
=A 1+B 1	=A 4+B 4
=A\$ 1+B 1	=A\$ 1+B 4
=SUM(A 1:B 1)	=SUM(A 4:B 4)

レコードの A 列の合計値などをフッタとして表示させる場合、セル型を文字列に設定して以下のように記述すると式として判断されます。



#### 出力しないシートについて

EGSheet#setHidden(hidden)を使用することでシートを出力しないようにすることができます。

true       : 出力しない  
false      : 出力する (デフォルト)

#### シートのレイアウト設定について

##### ● ウィンドウ枠の固定

EGSheet#setFixWindow(row, col)を使用することでウィンドウ枠の固定を設定することができます。

row           : 行番号  
col           : 列番号

##### ● 行高さの設定

EGRange#setHeight(row, height)を使用することで行の高さを個別に設定することができます。

row           : レコード内の行番号  
height       : 高さ (ピクセル)

## チュートリアル

ここでは ExcelGenerator を使用した Web アプリケーションの作成を説明していきます。

### 仕様

今回のチュートリアルで扱う Web アプリケーションはゴルフスコアデーター一覧のエクセルファイルを生成するサプレットを作成します。

開発・運用環境は下記の環境を想定します。

OS : Windows 2000

AS : Tomcat 4.1.24

JDK : 1.4.1\_03

ゴルフスコアデーター一覧ではあるゴルフ部に所属しているメンバーのスコア一覧を表示します。表示されるデータは名前、大会名、コース名、OUT スコア、IN スコア、GROSS スコア、ハンディキャップ、NET スコア、順位をそれぞれ表示します。

OUT スコア、IN スコアの合計から GROSS を算出します。

GROSS スコア、ハンディキャップから NET スコアを算出します。

計算部分はエクセルの計算機能を使用します。

### 雛型エクセルの準備

最終出力としては下記のような表を考えます。

日付: 平成16年1月19日

第3ゴルフ部スコア一覧表

名前	大会名	コース名	OUTスコア	INスコア	GROSSスコア	ハンディキャップ	NETスコア	順位
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1
富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	36	118	2
茄子 花子	富士ゴルフオープン	富士グランドクラブ	54	53	107	10	97	1

Copyright DaigoGolfCulb 2004

CopyRight Dai3GolfClub 2004

これをもとにヘッダ部、レコード部、フッタ部となる部分を考えます。

ヘッダ部は上部で固定となる部分なので下記のようなエクセルファイルを Golf\_H.xls として用意します。B1 セルは日付型に設定します。

	A	B	C	D	E	F	G	H	I
1	日付:	第3ゴルフ部スコア一覧表							
2									
3	名前	大会名	コース名	OUTスコア	INスコア	GROSSスコア	ハンディキャップ	NETスコア	順位
4									

レコード部は繰り返される部分なので下記のようなエクセルファイルを Golt\_R.xls として用意します。

	A	B	C	D	E	F	G	H	I
1									
2									

フッタ部は下部で固定となる部分なので下記のようなエクセルファイルを Golf\_F.xls として用意します。

	A	B	C	D	E	F	G	H	I
1									
2									
3									

## プログラム

サーブレットは GolfServlet.java という形で用意します。コードは下記のようになります。

-----ここから-----

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Calendar;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import jp.co.ffc.gkit.excelgenerator.EGBook;
import jp.co.ffc.gkit.excelgenerator.EGRange;
import jp.co.ffc.gkit.excelgenerator.EGSheet;

public class GolfServlet extends HttpServlet {
```



```

protected void doGet(
    HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {

    //コンテンツタイプを設定する
    response.setContentType("application/vnd.ms-excel;");

    //雛型ファイルのInputStreamを取得する
    InputStream ish =
        this.getServletContext().getResourceAsStream(
            "WEB-INF/tmp/sample/Golf_H.xls");
    InputStream isr =
        this.getServletContext().getResourceAsStream(
            "WEB-INF/tmp/sample/Golf_R.xls");
    InputStream isf =
        this.getServletContext().getResourceAsStream(
            "WEB-INF/tmp/sample/Golf_F.xls");

    //EGBookのインスタンスを生成する
    EGBook book = new EGBook(ish, isr, isf);

    //InputStreamをクローズする
    ish.close();
    isr.close();
    isf.close();

    //操作するシートを取得する
    EGSheet sheet = book.getSheetAt(0);

    //雛型ファイルの使用部分の行数を設定します。
    sheet.setHeaderRangeNum(3);
    sheet.setRecordRangeNum(1);
    sheet.setFooterRangeNum(2);

    //ヘッダの編集
    EGRange header = sheet.getHeaderRange();
    //今日の日付を入力する
    header.setData(0, 1, Calendar.getInstance());

```

```

//レコードの編集
for (int i = 0; i < 20; i++) {
    //新しいレコード部を取得する
    EGRange record = sheet.nextRecordRange();
    //レコードにデータを入力する
    record.setData(0, 0, "富士 太郎");
    record.setData(0, 1, "富士ゴルフオープン");
    record.setData(0, 2, "富士グランドクラブ");
    record.setData(0, 3, 79);
    record.setData(0, 4, 75);
    //式の代入
    record.setFormula(0,5,"D" + (i + 4) + "+E" + (i + 4));
    record.setData(0, 6, 35);
    record.setFormula(0, 7, "F" + (i + 4) + "-G" + (i + 4));
    record.setData(0, 8, 3);
}

//フッタの編集
EGRange footer = sheet.getFooterRange();
footer.setData(1,8,"CopyRight Dai3GolfCulb");

//出力先のOutputStreamを取得する
OutputStream output = response.getOutputStream();

//書き出す
book.write(output);
}
}

```

-----ここまで-----

#### 実行

Tomcat を実行し <http://localhost:8080/EXCELGENERATOR/servlet/GolfServlet> にアクセスすると下記のような表示になりデータを入力されたエクセルファイルが表示されます。

サンプル - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) 移動(Q) お気に入り(A) ヘルプ(H)

戻る 検索 お気に入り メディア

G32

	A	B	C	D	E	F	G	H	I
1	日付:	平成15年12月19日							
2	<b>第3ゴルフ部スコア一覧表</b>								
3	名前	大会名	コース名	OUTスコア	INスコア	GROSSスコア	ハンディキャップ	NETスコア	順位
4	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
5	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
6	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
7	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
8	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
9	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
10	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
11	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
12	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
13	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
14	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
15	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
16	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
17	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
18	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
19	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
20	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
21	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
22	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
23	富士 太郎	富士ゴルフオープン	富士グランドクラブ	79	75	154	35	119	3
24									
25									
26									
27									

Copyright Dai3GolfClub

## HOW TO

ここではテクニックや問題解決手段を提示します。参考にしてください。

### 雛型シートの内容が表示されない

EGSheet# setHeaderRangeNum(int)、EGSheet# setRecordRangeNum(int)、EGSheet# setFooterRangeNum(int)を設定しているか確認してください。それぞれのレンジの表示行数は初期値0となっています。それぞれのレンジとして使用する行数を設定してください。

### レコード部またはフッタ部が必要ない

レコード部、フッタ部が必要ないときは EGBook のコンストラクタ EGBook(InputStream, InputStream, InputStream)の第二引数、第三引数に null を指定してください。第二引数はレコードの雛型エクセルの InputStream、第三引数はフッタの雛型エクセルの InputStream となります。ただし、ヘッダ部となる第一引数に null を指定することはできません。

### 出力されるシート数をプログラムで増やす

データ量によりシート数が増えるような帳票を作成する際は EGBook# copySheet(int)を使用してください。引数で指定した番号のシートのコピーが最終シートに挿入されます。その際、プログ

ラムで設定したデータはコピーされませんので注意してください。

#### 印刷範囲を 1 枚に収めるには？

印刷範囲を 1 枚に収めるには EGSheet#setFitHeight(short)、EGSheet#setFitWidth(short) を使用して印刷の縦横枚数を 1 に設定してください。

#### サブレットにブラウザでアクセスしてもエクセルが開かない

クライアントの環境によってはブラウザでファイルが開かないことがあります。下記の項目を確認してください。

- ・ Internet Explorer 5.5 以上を使用しているか？
- ・ Excel 97 または Excel 2000 以上がインストールされているか？

もしクライアントに問題がないときはサーバ側設定の問題があります。

- ・ コンテンツタイプを "application/vnd.ms-excel" に設定しているか？
- ・ サーバ側でエラーが表示されていないか？

#### 雛型エクセルファイルに設定した内容が出力エクセルファイルに反映されない

雛型エクセルファイルの設定内容（罫線や背景色等）が出力エクセルファイルに反映されないことがあります。このような場合は再度、雛型エクセルファイルの設定を見直してください。

- ・ 最上行の上罫線、最左列の左罫線等は設定を確認する際、セルの書式設定で確認するようにしてください。
- ・ シート全体に設定がしてある場合は ExcelGenerator が認識することができません。そのような場合は必要個所のみに設定を行うようにしてください。

#### 印刷ヘッダ・フッタを付けたい

印刷ヘッダ・フッタをつけることはできません。シート内に印刷ヘッダ・フッタが表示されるよう設定を行ってください。

#### 開発中にエクセルで雛型ファイルが開かなくなるときがある

雛型エクセルを修正する為にエクセルを開こうとしても開かない場合はサブレットプログラム中で InputStream をクローズしていない可能性があります。EGBook のインスタンスを生成した後に使用していた InputStream をクローズしてください。

InputStream#close()

## Javadoc

ExcelGenerator の Javadoc はインストールフォルダにありますので、そちらを参照ください。

## ライセンス

ExcelGenerator のライセンスおよび Jakarta POI のライセンスについて説明いたします。

### Jakarta POI ライセンス

ExcelGenerator はエクセル生成に関して Jakarta POI を使用しております。Jakarta POI のライセンスは下記のようにしております。

<http://www.apache.org/licenses/LICENSE>

```
/* =====
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 * if any, must include the following acknowledgment:
 *
 * "This product includes software developed by the
 * Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software itself,
 * if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation" must
 * not be used to endorse or promote products derived from this
 * software without prior written permission. For written
 * permission, please contact apache@apache.org.
```

\*  
 \* 5. Products derived from this software may not be called "Apache",  
 \* nor may "Apache" appear in their name, without prior written  
 \* permission of the Apache Software Foundation.  
 \*  
 \* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED  
 \* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
 \* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
 \* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR  
 \* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
 \* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT  
 \* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF  
 \* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND  
 \* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT  
 \* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
 \* SUCH DAMAGE.  
 \*  
 \* =====  
 \*  
 \* This software consists of voluntary contributions made by many  
 \* individuals on behalf of the Apache Software Foundation. For more  
 \* information on the Apache Software Foundation, please see  
 \* <<http://www.apache.org/>>.  
 \*  
 \* Portions of this software are based upon public domain software  
 \* originally written at the National Center for Supercomputing Applications,  
 \* University of Illinois, Urbana-Champaign.  
 \*/

## ExcelGenerator ライセンス

ExcelGenerator のライセンス形態についてご説明いたします。ExcelGenerator は CPU ライセンスとなっております。運用されるサーバの CPU 1 個に対し 1 つのライセンスが必要になります。

製品名	価格	型番
GkitTaglib ExcelGenerator	175,000 円 (税込み 183,750 円)	A2091393-81
1 ライセンス	140,000 円 (税込み 147,000 円)	A2091393-82
5 ライセンス	673,000 円 (税込み 706,650 円)	A2091393-83
20 ライセンス	2,625,000 円 (税込み 2,756,250 円)	A2091393-84

上記の価格表は 2003 年 1 月 19 日現在のものです。状況により価格が変動することがありますので、ご購入時の価格については製品ホームページにてご確認ください。

<http://www.ffc.co.jp/products/gkittaglib/index.html>

開発ライセンスはフリーです。また、Web アプリケーションサーバでの動作を想定しておりますが、Java アプリケーションとしてクライアントで動作させる場合のランタイムライセンスは別途製品サポートまでご連絡ください。弊社営業より追ってご連絡いたします。

## サポートについて

製品のサポートについて記述しております。お問合せの際は下記の内容を一読してお問合せください。

### 無償サポート

製品の問い合わせは基本的に無償となっております。サポートのやり取りはすべて電子メールまたは FAX にて対応させていただいております。ご質問内容や状況によっては早急に回答できないことがあります。予めご了承ください。

電子メール：[gkit@ml.ffc.co.jp](mailto:gkit@ml.ffc.co.jp)

FAX 番号：03-5324-1667（GKitTaglib ExcelGenerator と明記してください）

お問合せの際は以下の内容にてお問合せいただきますようお願いいたします。

- ・ **タイトル**：お問い合わせの内容が判断できるような簡潔なものとしてください。
- ・ **連絡先**：お客様の会社名、氏名、メールアドレス、電話番号、FAX 番号
- ・ **製品名**：お問い合わせの製品名を記載してください。
- ・ **開発環境**：OS、AS、JDKの製品名およびバージョン
- ・ **運用環境**：OS、AS、JDKの製品名およびバージョン
- ・ **内容**：お問い合わせの内容を詳細に記載してください。

障害と思われる現象についてはどのような操作をしたときに発生するのか、また発生頻度（常に発生するのか、特定の条件でのみ発生するのかなど）も必ず記載してください。

可能であれば、スタンドアロンでの現象が確認できる再現プログラムをお送りください。再現確認の時間が短縮されることによって、より早い解決に繋がります。

### 有償サポート

有償サポートについてはライセンス数により異なります。GKitTaglib 製品すべてが対象になります。例えば GKitTaglib Spreadsheet と GKitTaglib ExcelGenerator を 1 ライセンスご使用の場合は 2 ライセンスでのサポートとなります。有償サポートライセンスは下記を参照してください。

製品名	製品番号	価格			
		1 本	2 本	3 ～ 5 本	6 本以上
1 ヶ月サポートサービス	A2091393-41	150,000 円	225,000 円	300,000 円	450,000 円

3 ヶ月サポートサービス	A2091393-42	400,000 円	600,000 円	800,000 円	1,200,000 円
6 ヶ月サポートサービス	A2091393-43	750,000 円	1,125,000 円	1,500,000 円	2,250,000 円
12 ヶ月サポートサービス	A2091393-44	1,300,000 円	1,950,000 円	2,600,000 円	3,900,000 円

#### 【無償・有償サポートの違い】

##### ( 1 ) 技術質問・障害調査等の優先対応

弊社技術者がお客様の質問に対し優先的に対応します。サンプル作成等も行います。

##### ( 2 ) 専用メールアドレスの作成

お客様専用の問い合わせ用のメールアドレスを作成し、営業日の 17:00 までに受け付けた内容を翌営業日中に回答します

##### ( 3 ) 問い合わせ回数は無制限

期間中であれば無制限に問い合わせを行うことができます。

##### ( 4 ) レベルアップサービス

サポート期間中にモジュールのレベルアップがあった場合、メールにてご連絡いたします。

有償サポートについての詳細な内容は弊社サポートにお問合せください。

### 制限事項

ExcelGenerator の制限事項について説明いたします。

- ・印刷ヘッダ、印刷フッタを設定することはできません。
- ・同じセル内での異なるフォント設定はサポートしておりません。
- ・雛型エクセルはシート全体に設定を行うと出力エクセルに反映することができません。フォントや背景色や罫線の設定はセルごとに行うようにしてください。
- ・雛型エクセルの同一セル内で異なるフォント設定が行われていた場合は出力エクセルに反映することはできません。
- ・マクロを使用したエクセルファイルを雛型として使用することはできません。
- ・図やオブジェクト等を配置したエクセルを雛型として使用することはできません。
- ・グラフを用いたエクセルファイルを雛型として使用することはできません。
- ・セルの設定を縦書きに設定したエクセルファイルを雛型としてしようすることはできません。



- ・ 全角文字が含まれているセルのフォーマットは設定することはできません。
- ・ フリガナには対応していません。
- ・ セルの斜め罫線には対応していません。
- ・ 機能一覧の雛型に×のついた項目については雛型エクセルファイルとして設定することはできません。

