

コードレビュー

Vol.21

USP 研究所技術研究員
written by 大内智明

CODE Review

業務で使用するエクセルファイルの作成について説明します。

業務で使用するエクセルファイル

企業では、様々な業務が存在して、1つ1つがIT化により多種多様なデータを持っています。そうしたサーバに入っているデータは、そのままの状態だと簡単に加工や編集、表示ができません。人が業務サーバ内にあるデータを整理して見るツールとして、HTML・PDF・CSV・EXCELがあります。今回は、帳票・マスタ関連情報・在庫情報といったデータ出力するのに、ツールの中でも誰でも操作ができるエクセルを使用してみます。

今回のコードレビューは、テキストデータからエクセルファイルの作成方法について説明します。

技術的な概要

ユニケーシでは、wexcel コマンドを使用して、データが入力されたテキストデータを予め準備したエクセルのテンプレートに書き込むことで、エクセルファイルを作成します。

図1 エクセルファイルを出力

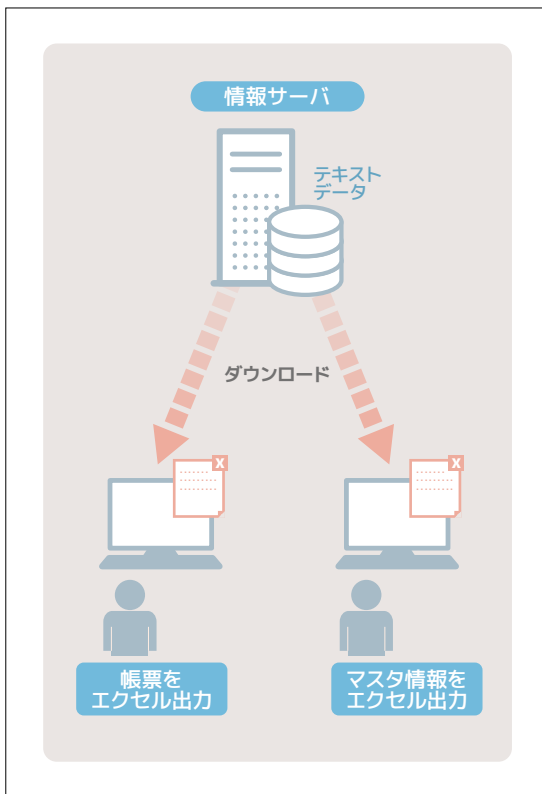


図2 エクセルの作成

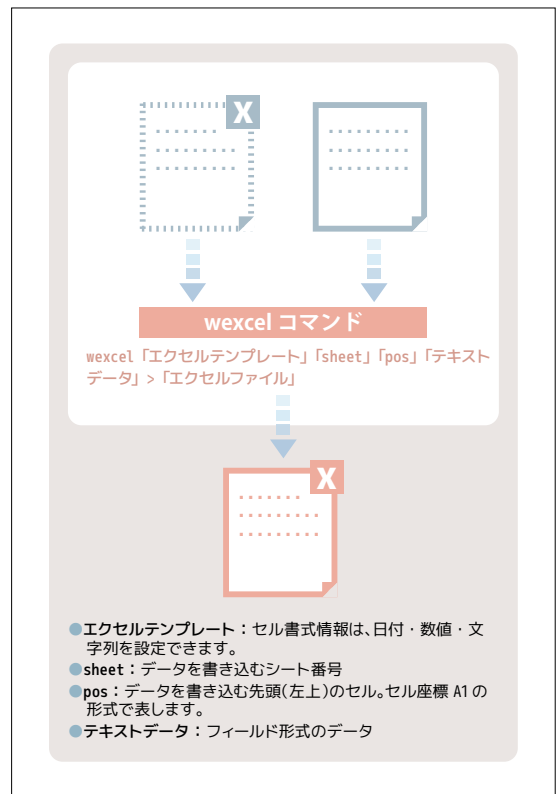


図3 テキストデータを書き込んだエクセルファイル

NO	処理区分	適用日	状態	店舗基本情報							店舗住所情報
				店舗コード	事業会社 コード	店舗名(漢字)	店舗名(カナ)	店舗短縮名	店舗区分	売価優先 区分	郵便番号
				表示							
	1:追加 2:変更 3:削除	{yyyymmdd} ※処理区分が「1 (追加)」の場合 は、当日	有効 削除	半角数字6桁	半角数字3桁	全角25文字以内	全角25文字以内	全角25文字以内	1:直営 2:FC 4:DC	0:無し 1:売価優先	ハイフン無しの 半角数字7桁
1			有効	000001	001	品川店	シナガワテン	品川	1	0	1112222
2			有効	000002	001	新宿店	シンジュクテン	新宿	1	0	1113333
3			有効	000003	001	仙台店	センダイテン	仙台	1	0	1114444
4			有効	000004	001	浜松店	ハママツテン	浜松	1	0	1115555
5											
6											
7											
8											
9											
10											

リスト エクセルを作成するシェル

```

1 #!/bin/ush -xve
2 # システム名      : U S Pシステム
3 # サブシステム名: 店舗メンテナンス
4 # 作成日          : 20xx/xx/xx
5 # 会社名          : USP
6 # 作成者          : xxxx
7
8 #####
9 # 初期設定
10 #####
11
12 #-----
13 # 変数の定義
14 #-----
15 HOME=/home/usp
16 tmp="/tmp/${$-(basename $0)}_$(date +%Y%m%d%H%M%S)" # 一時ファイル
17
18 apld=${HOME}/AP/TENPO # 本アプリケーションのディレクトリ
19 lv4d=${apld}/LV4/TENPO # LV4データ用ディレクトリのパス
20 lv3d=/home/DATA/LV3 # LV3データ用ディレクトリのパス
21
22 # エラー時の終了処理定義 (システムエラー系)
23 err ERROR_EXIT(){
24 # CGI ヘエラーで返す
25 exit 1
26 }
27
28 # 引数の確認
29 [ $# -ne 3 ] && ERROR_EXIT
30
31 flddata=$1
32 sdaytime=$2
33 ppid=$3
34
35 # 入力ファイル有無チェック
36 [ ! -s ${flddata} ] && ERROR_EXIT
37
38 #####
39 # EXCELテンプレートのパスの取得
40 #####
41
42 # EXCELテンプレートの絶対パス(ファイル名付き)
43 excel_tmplate=${apld}/TEMPLATE/0000.xls
44
45 # 保存ファイル名の作成("書式ID_書式表示名_CGI実行日時.xls")

```

```

46 excel_savename=excel_out.xls
47
48 #/////////////////////////////////////////////////////////////////
49 # 共通パラメータの取得
50 #/////////////////////////////////////////////////////////////////
51
52 # APディレクトリ名を設定する
53 aplid="TENPO"
54
55 # サーバに設定された業務日付を取得する
56 gday=$(date+%Y%m%d)
57 # 画面からの共通パラメータにより共通チェック用の値を取得
58 # 1:MD業務グループコード 2:店舗コード 3:ユーザID
59 # ユーザID
60 userid=$(head -1 ${flddata} | self 3)
61 # MD業務グループコード
62 mdgrp=$(head -1 ${flddata} | self 1 )
63 # 対象店舗コード(任意条件)
64 lineup 2 ${flddata} |
65 delr 1 " " > ${tmp}-taishotenpocd
66 # 1:店舗コード
67
68 # MD業務グループに対応する事業会社コードを取得
69 selr 2 ${mdgrp} ${lv3d}/SONOTA /TBL/JGYKCD_MDGYOMUGRP |
70 # 1:事業会社コード 2:MD業務グループ 3:更新フラグ 4:適用日
71 # 5:更新者 6:更新日時
72 lineup 1 > ${tmp}-jgyocd
73
74 # 対象事業会社コードを取得
75 # 無印L V 3を直接参照
76 # ファイルを保証
77 touch ${tmp}-tantojgyo
78 if [ -s ${lv3d}/JUGYOUIN_JGYOK/TBL/JUGYOUIN_JGYOK ] ; then
79 cat ${lv3d}/JUGYOUIN_JGYOK/TBL/JUGYOUIN_JGYOK |
80 # 1:ログインID* 2:担当事業会社コード* 3:適用日*
81 # MD業務グループに対応した事業会社コードで絞る
82 cjoin0 key=2 ${tmp}-jgyocd - |
83 selr 1 ${userid} |
84 # 事業会社コードをユニーク化
85 lineup 2 > ${tmp}-tantojgyo
86 # 1:担当事業会社コード
87 fi
88
89 #-----
90 # ディレクトリ整備
91 #-----
92 # ダウンロード用ディレクトリ(テンポラリ用)
93 dwnd=/home/usp/AP/DOWNLOAD/TMP/${gday}
94
95 # 出力EXCELファイル名
96 outexcel=${dwnd}/${sdaytime}.${userid}.${ppid}.${basename ${excel_tmplate}}
97
98 # ディレクトリ作成
99 mkdir -p ${dwnd}
100
101 #/////////////////////////////////////////////////////////////////
102 # 処理部
103 #/////////////////////////////////////////////////////////////////

```

指定したフィールドのデータのラインナップを取り出す

```

$ cat data
1 1 717300 abc1
1 2 717301 abc2
1 3 717302 abc3
1 1 717303 abc4
1 2 717304 abc5
$ lineup 2 data
1
2
3

```

delrはユニーク化コマンド、画面1参照

selrはユニーク化コマンド、画面2参照

シーケンシャルキーマッチング
セレクト

```

$ cat master
0000003 杉山_____ 26 F
0000005 崎村_____ 50 F
0000007 梶川_____ 42 F
$ cat tran
0000005 82 79 16 21 80
0000001 46 39 8 5 21
0000004 58 71 20 10 6
0000009 60 89 33 18 6
0000003 30 50 71 36 30
0000007 50 2 33 15 62
$ cjoin0 key=1 master tran > ok
$ cat ok
0000005 82 79 16 21 80
0000003 30 50 71 36 30
0000007 50 2 33 15 62

```

```

104 #-----
105 # LV4ファイル作成
106 #-----
107 #-----
108 # 基本項目
109 #-----
110 # 出力ファイルの初期化
111 touch ${tmp}-kihon
112
113 # LV4の存在チェック
114 if [ -s ${lv4d}/TENPO_KIHON ] ; then
115     # 1:事業会社コード 2:店舗コード 3:店舗名称 4:店舗名称(カナ)
116     # 5:店舗短縮名 6:店舗区分 7:売価優先区分 8:郵便番号
117     # 9:都道府県コード 10:住所1 11:住所2 12:住所3
118     # 13:電話番号1 14:電話番号2 15:短縮電話番号 16:FAX番号1
119     # 17:FAX番号2 18:店舗写真フラグ 19:適用日 20:更新フラグ
120     # 21:ユーザID 22:ユーザ名 23:コンピュータ名 24:削除フラグ
121     # 25:更新フラグ 26:適用日 27:更新者 28:更新日時
122     # 履歴検索
123     cat ${lv4d}/TENPO_KIHON |
124     # 重複するレコードを1レコードにする(有効データの項目が残る)
125     self 2 1 3/NF |
126     # 対象店舗コード(任意条件)で絞り込み
127     if [ -s ${tmp}-taishotenpocd ] ; then
128         cjoin0 key=1 ${tmp}-taishotenpocd -
129     else
130         cat -
131     fi
132     # 1:店舗コード 2:事業会社コード 3:店舗名称 4:店舗名称(カナ)
133     # 5:店舗短縮名 6:店舗区分 7:売価優先区分 8:郵便番号
134     # 9:都道府県コード 10:住所1 11:住所2 12:住所3
135     # 13:電話番号1 14:電話番号2 15:短縮電話番号 16:FAX番号1
136     # 17:FAX番号2 18:店舗写真フラグ 19:適用日 20:更新フラグ
137     # 21:ユーザID 22:ユーザ名 23:コンピュータ名 24:削除フラグ
138     # 25:更新フラグ 26:適用日 27:更新者 28:更新日時
139     LANG=C sort -k1,1 -k26,26 -k28,28 |
140     # 担当事業会社コードで絞り込み
141     cjoin0 key=2 ${tmp}-tantojgyo -
142     fi
143
144 # 並び替え
145 msort key=1
146 self 1/8 10/17
147 # 1:状態フラグ(仮) 2:店舗コード 3:事業会社コード 4:店舗名(漢字)
148 # 5:店舗名(カナ) 6:店舗短縮名 7:店舗区分 8:売価優先区分
149 # 9:郵便番号 10:住所1(都道府県) 11:住所2(市町村) 12:住所3(丁目等)
150 # 13:電話番号1 14:電話番号2 15:短縮電話番号 16:FAX番号1
151 # 17:FAX番号2 18:開店日 19:閉店日 20:発注開始日
152 awk '{print " _ 有効", $0}' |
153 rank > ${tmp}-excel-out
154 # 1:通番 2:処理区分 3:適用日 4:状態フラグ(仮)
155 # 5:店舗コード 6:事業会社コード 7:店舗名(漢字) 8:店舗名(カナ)
156 # 9:店舗短縮名 10:店舗区分 11:売価優先区分 12:郵便番号
157 # 13:住所1(都道府県) 14:住所2(市町村) 15:住所3(丁目等) 16:電話番号1
158 # 17:電話番号2 18:短縮電話番号 19:FAX番号1 20:FAX番号2
159
160 #/////////////////////////////////////////////////////////////////
161 # Excelテンプレートにデータをセット

```

ランクを付加する。引数を何も指定しないときは、単純な行番号を付加する

```

$ cat data
JPN 杉山_____ 26
JPN 崎村_____ 27
JPN 梶川_____ 27

$ rank data
1 JPN 杉山_____ 26
2 JPN 崎村_____ 27
3 JPN 梶川_____ 27

```

msort はユニケージコマンド。
オンメモリソート。
sort コマンドの高速版。

```

162 #/////////////////////////////////////////////////////////////////
163 if [ -s ${tmp}-excel-out ]; then .....
164
165 wexcel -i_ -d ${excel_tmplate} \
166     1 a11 ${tmp}-excel-out \
167     2> ${tmp}-wexcel-stderr > ${tmp}-exceldata
168
169 # wexcel のエラーをチェック
170 if [ -s ${tmp}-wexcel-stderr ]; then
171     ERROR_EXIT
172 fi
173
174 else
175     # 検索結果が0件の場合は、テンプレートをそのまま返す
176     cp ${excel_tmplate} ${tmp}-exceldata
177
178 fi
179
180 # 作成したEXCELを保存
181 mv ${tmp}-exceldata ${outexcel} .....
182
183
184 #/////////////////////////////////////////////////////////////////
185 # 終了
186 #/////////////////////////////////////////////////////////////////
187 rm -f ${tmp}.* > /dev/null
188 exit 0

```

wexcel はユニケージコマンド
エクセル帳票テンプレートに
データをはめ込む。使用方法
は、「技術的な概論」を参照。

画面1 delr

フィールドの非完全一致を見る

```

$ cat data
0001 a
0002 b
0003 c
0004 c
$ delr 1 "0001" < data
0002 b
0003 c
0004 c

```

画面2 selr

フィールドの完全一致を見る

```

$ cat data
0001 a
0002 b
0003 c
0004 c
$ selr 1 "0001" < data
0001 a

```

コードの見どころ

- [1] LV4データを加工します (① 110 ~158行目)。
- [2] 加工したテキストデータをエクセルテンプレートに書き込みして、エクセルファイルを作成します (② 163 ~181行目)。

まとめ

業務で使用しているサーバ内のデータは、複数ファイルに分かれて存在しています。今回は、それらを集計・加工して、テキストデータを作成しました。次に作成したテキストデータは、定型のエクセルファイルの開始 xy 座標を指定して、1 コマンドではめ込み、エクセルデータ出力の説明を行いました。

次回は、エクセルファイルの読み込みについて説明します。