

## データ編集

|                          |    |
|--------------------------|----|
| 1. 概要 .....              | 2  |
| 2. 設定方法 .....            | 3  |
| 2-1. データ編集項目の作成と登録 ..... | 3  |
| 2-2. データ編集項目の設定 .....    | 6  |
| 3. 標準コマンド .....          | 8  |
| 3-1. 元号対応 .....          | 8  |
| 4. フリーフォーマットコマンド .....   | 10 |
| 4-1. ins コマンド .....      | 10 |
| 4-2. rep コマンド .....      | 12 |
| 4-3. cnv コマンド .....      | 13 |
| 4-4. ded コマンド .....      | 14 |
| 4-5. adj コマンド .....      | 15 |
| 4-6. gen コマンド .....      | 16 |
| 4-7. ext コマンド .....      | 16 |
| 4-8. コマンドの複数指定 .....     | 17 |

## 1. 概要

データ編集とは、入力されたデータに対して何らかの加工を行う機能です。

例えば、金額のデータ“1234”が入力された場合、この金額のデータにカンマや通貨記号を付けて“¥1,234”という形で出力することができます。これがデータ編集の機能です。

なお、データ編集は変数オブジェクト（項目データ）に対して予め用意されたデータ編集の形式を選択して指定します。

データ編集の指定方法としては、以下の形式があります。

- ・ 金額、日付、曜日、時刻などの標準のデータ編集  
通貨記号を付加したり、西暦を和暦に変更するといった標準のデータ編集です。
- ・ フリーフォーマットコマンドによるデータ編集  
予め用意したデータ編集用のコマンドにより、ユーザ独自の柔軟なデータの編集が可能です。
- ・ dff ファイルによるデータ編集  
dff ファイルへフリーフォーマットコマンドを記述し、この dff ファイルをデータ編集に利用します。

## 2. 設定方法

データ編集の設定は、Form エディタから行います。

なお、標準で登録されているデータ編集以外を使用する場合や、フリーフォーマットコマンドを使用する場合は、データ編集項目を作成してデータ編集登録を行う必要があります。

### 2-1. データ編集項目の作成と登録

例として、数値を小数 3 桁に四捨五入するデータ編集項目の作成方法について説明します。

Form エディタからデータ編集登録ダイアログを起動します。

#### ◆◆ 操作 ◆◆

Form エディタを起動し、ツールボタン [ データ編集登録 ] をクリックします。  
(ツールメニュー [ 編集 ] - [ データ編集登録 ] を選択しても同様です。)

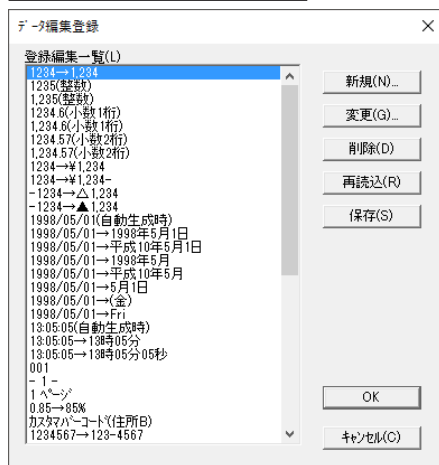
図：ツールボタン [ データ編集登録 ]



データ編集登録ダイアログが表示されます。

左の登録編集一覧のリストに並んでいるものが、現在登録されているデータ編集項目です。

図：データ編集登録ダイアログ



このデータ編集登録ダイアログから、数値を小数 3 桁に四捨五入するデータ編集を作成します。

#### ◆◆ 操作 ◆◆

[ 新規 ] ボタンをクリックします。

図：データ編集作成ダイアログ

## ◆◆ 操作 ◆◆

[編集種別] から [数値] を選択し、[数値の処理] で [桁] にチェックを入れ、[小数3桁] を選択します。

この際、[変換サンプル] で変換元の数値を変更すると、変更後の値が現在の設定に合わせて変換されます。

この変換結果を参考にして、データ編集の作成を行うことができます。

## ◆◆ 操作 ◆◆

[タイトル文字列] に現在のデータ編集設定を保存するための任意の名前を入力します。  
ここでは、「小数3桁 (サンプル)」と入力します。

図：データ編集作成ダイアログ

データ編集作成 - 数値

編集種別(G) 基本パターン(P)

数値  
金額  
日付  
曜日  
時刻  
ページ  
割合  
%  
フリー

☐ 桁区切り(C)を付加

全角/半角(B)

☒ 元のデータに依存  
☐ 強制的に全角で出力  
☐ 強制的に半角で出力

数値の処理(Y)

☒ 桁 小数3桁  
 丸め処理 四捨五入

変換サンプル(S)

変換元 1234.5678 → 変換後 1234.568

タイトル文字列(T) 少数3桁(サンプル)

OK キャンセル(C)

## ◆ ◆ 操作 ◆ ◆

[OK] ボタンをクリックし、データ編集作成ダイアログを終了します。

データ編集登録ダイアログのリストに、追加したデータ編集項目が登録されていることを確認し、[保存] ボタンをクリックします。

最後に [OK] ボタンでダイアログを終了します。

図：データ編集登録ダイアログ

データ編集登録

登録編集一覧(L)

001  
1  
1ページ  
0.95→95%  
カスタマバート(住所B)  
1234567→123-4567  
100-0001→〒100-0001  
1234567→〒123-4567  
値→(個)スペースは削除  
3→■■■  
1,234円  
10→10件  
1→「○」0→「」  
123 → 123.00(桁揃え)  
1234 → 1,234.00(少数2桁揃え)  
1234 → 1,234.000(少数3桁揃え)  
少数2桁揃え 0→削除 カマ編集無  
少数2桁揃え 0→削除 カマ編集有  
1 → 1(10桁揃え)  
1 → 0000000000(10桁揃え)  
1234 → ( 1,234 ) 1桁揃え  
1234 → (1,234) カマ2カマ編集  
2012/01/02 → 2012年01月02日  
2012/01/02 → 平成24年01月02日  
○○○→○○○ 御中  
○○○→○○○ 様  
30 → 30歳  
少数3桁(サンプル)

新規(N)...  
変更(G)...  
削除(D)  
再読み込み(R)  
保存(S)

OK  
キャンセル(C)

以上でデータ編集項目が登録されました。

これで、Form エディタ上から追加したデータ編集項目を利用することができます。

## 2-2. データ編集項目の設定

データ編集項目の設定方法について説明します。

例として、「2-1. データ編集の作成」で作成したデータ編集項目をオブジェクトに設定します。

Formエディタ上でデータ編集項目を設定したいオブジェクトを選択して、プロパティダイアログを開きます。

### ◆◆ 操作 ◆◆

オブジェクトをダブルクリックします。

(オブジェクトを右クリックして[コンテキストメニュー]-[プロパティ]をクリックしても同様です。)

プロパティダイアログが起動しますので、データ編集項目を設定します。

### ◆◆ 操作 ◆◆

プロパティダイアログにおいて、[編集]タブをクリックします。

[データ編集]コンボボックスから「少数3桁(サンプル)」をクリックして選択します。

図：変数オブジェクトのプロパティダイアログ

図：表オブジェクト（セル）のプロパティダイアログ

セルの設定 [レコードブロック]

セル名: ITEM001

形式: テキスト

属性 | フォント | 配置 | 編集 | 罫線

データ編集: 少数3桁(サンプル)

☐ 領域外のデータを削除する

適用 OK キャンセル

設定内容を保存します。

◆◆ 操作 ◆◆

[OK] ボタンをクリックします。

以上でデータ編集項目が設定されました。

### 3. 標準コマンド

設定可能なデータ編集の標準コマンドは以下の通りです。

- 数値 : 数値の桁区切り、丸めなどを設定できます。
- 金額 : 数値に加え、通貨記号、負数表示などを設定できます。
- 日付 : 日付配置の並び、区切りスタイルなどを設定できます。
- 曜日 : 漢字・英字出力などを設定できます。
- 時刻 : 時刻配置の並び、区切りスタイルなどを設定できます。
- ページ : 桁数、ページを括る付加表示などを設定できます。
- 割合 : 割 - 分 - 厘の付加などを設定できます。
- % : 桁数、単位付加などを設定できます。
- フリー : フリーフォーマットコマンドを用いて、ユーザ独自のデータ編集項目を作成できます。dff ファイルによるデータ編集も、このフリーを使用します。

#### 3-1. 元号対応

データ編集コマンド「日付」設定において、和暦の「令和」以降の元号を表示する場合、以下の場所に元号設定ファイル「era\_jp.dat」ファイルを作成、または修正する必要があります。

Windows 環境 : ユーザ設定（共通）フォルダ

Linux/UNIX 環境 : ランタイム製品導入ディレクトリ

「era\_jp.dat」ファイルの詳細に関してはマネージャのメニュー [ ヘルプ ] - [ オンラインマニュアル ] から「1. インストール」-「1.6 導入環境の設定」-「1-6-9 元号対応」をご覧ください。

「era\_jp.dat」ファイルを用いた元号の設定例

ex1) era\_jp.dat ファイルに追加された元号を和暦（漢字）で表示する

era\_jp.dat ファイルの記述

2019/05/01, 令和, 令, R

データ編集設定

編集種別 : 日付

配置の並び : 年 - 月 - 日

区切りスタイル : 漢字 ( " 年", " 月", " 日" )

西暦 / 和暦 : 和暦表示

年の付加表示 : 先頭に元号 (漢字)

元年 / 1 年 : 元年

変換元 : 20190501

変換後 : 令和元年 5 月 1 日

ex2) era\_jp. dat ファイルに追加された元号を和暦（英大文字 1 字）で表示する  
era\_jp. dat ファイルの記述

2019/05/01, 令和, 令, R  
# 元号を追加しました  
2021/09/01, 久里, 久, C

#### データ編集設定

編集種別：日付

配置の並び：年－月－日

区切りスタイル：漢字（” 年”、” 月”、” 日”）

西暦 / 和暦：和暦表示

年の付加表示：先頭に元号（英大文字 1 字）

変換元：20231001

変換後：C3 年 10 月 1 日

## 4. フリーフォーマットコマンド

フリーフォーマットコマンドとは、予め用意したデータ編集用のコマンドから、ユーザ独自の柔軟なデータ編集を行うことができる機能です。

フリーフォーマットで使用できるコマンドは以下の通りです。

|            |                             |
|------------|-----------------------------|
| 挿入 (ins)   | : 対象データに文字列を付加・挿入する         |
| 置換 (rep)   | : 対象データの一部または全部を他の文字列に置き換える |
| 変換 (cnv)   | : 対象データの一部または全部の文字タイプを変換する  |
| 数値 (ded)   | : 半角数値データを編集する              |
| 字間 (adj)   | : 対象データの文字間隔を調整する           |
| 繰返 (gen)   | : 半角数値データを繰返し文字列に変換する       |
| ファイル (ext) | : dff ファイルからデータ編集を利用する      |

フリーフォーマット文字列は、以下のようにコマンドとパラメータをカンマで区切って指定します。

コマンド, パラメータ [, パラメータ, ...]

以下にコマンドごとのフォーマット文字列の記述方法を示します。

記号の意味:

|            |                   |
|------------|-------------------|
| <>:        | 必須設定              |
| []:        | 任意設定              |
| a / b / c: | a, b, cの中から複数選択可  |
| a   b   c: | a, b, cの中から一つだけ選択 |

### 4-1. ins コマンド

編集データにパラメータで指定した文字列を付加・挿入します。

**フォーマット文字列:**

```
ins [ , st = [ - ] n ] , "string"
ins < , st = [ - ] n / , ed = [ - ] n / , tc = [ - | +- ] "string" > , "string"
ins < , st = [ - ] n / , it = n / , rp = [ - ] n > , "string"
※ tc を指定した場合は it、rp の指定は無効になります。
```

st=[-]n: 始点の指定 (デフォルト値) st=0

n ≥ 0: データの先頭から n バイト目 (の後)

n < 0: データの末尾から n バイト目 (の前)

※ n = -0 を設定すると、末尾に文字列を挿入することができます。

※ n バイト目が全角文字の 2 バイト目の時はその全角文字が始点になります。

ed=[-]n : 終点の指定 (デフォルト値) ed=-0

n > 0 : データの先頭から n バイト目

n < 0 : データの末尾から n バイト目

※ n バイト目が全角文字の 2 バイト目の時はその全角文字が終点になります。

tc=[-|+]"string" : 文字列による挿入位置の指定

なし : "string" の後に挿入

- : "string" の前に挿入

+- : "string" の前後に挿入

it=n : 挿入間隔の指定 (デフォルト値) it=1

n : n バイトおきに挿入

rp=[-]n : 挿入回数の指定 (デフォルト値) rp=1

n > 0 : 右方向に n 回繰り返す

n < 0 : 左方向に n 回繰り返す

※挿入場所が全角文字の 1 バイト目と 2 バイト目の間だった場合はその文字の後に挿入します。

ex1) データの先頭に "Mr." を付加する

ins, "Mr."

変換元 : Suzuki

変換後 : Mr. Suzuki

ex2) データの末尾に "さん" を付加する

ins, st=-0, "さん"

変換元 : 鈴木紀子

変換後 : 鈴木紀子さん

ex3) データ (半角) の先頭から 5 バイト目に "\*" を付加する

ins, st=5, "\*"

変換元 : ABCDEFG

変換後 : ABCDE\*FG

ex4) データ (全角) の先頭から 5 バイト目に "\*" を付加する

ins, st=5, "\*"

変換元 : A B C D E F G

変換後 : A B C \* D E F G

ex5) データの先頭から 5 バイト目以降にある "C" の後に "\_" を挿入する

ins, st=5, tc="C", "\_"

変換元 : ABCABCABCABC

変換後 : ABCABC\_ABC\_ABC\_

ex6) データの先頭以降にある "-" の前後に " " (スペース) を挿入する

ins, st=0, tc="+-"," " " "

変換元 : 03-3456-7890

変換後 : 03 - 3456 - 7890

ex7) データの末尾から数えて 4 バイト目の前から、左方向に 4 バイトおきに “-” を 2 回挿入する

```
ins, st=-4, it=4, rp=-2, “-”
```

変換元 : 0334567890

変換後 : 03-3456-7890

## 4-2. rep コマンド

編集データの一部または全部を、パラメータで指定した文字列に置き換えます。

### フォーマット文字列：

```
rep [ , st = [ - ] n [ , pc = [ - ] “string” ] / , ed = [ - ] n [ , pc = [ - ] “string” ] / , tc = “string” ] , “string”
```

st=[-]n : 始点の指定 (n=0 は不可) (デフォルト値) st = 1

n > 0 : データの先頭から n バイト目

n < 0 : データの末尾から n バイト目

※ n バイト目が全角文字の 2 バイト目の時はその全角文字が始点になります。

ed=[-]n : 終点の指定 (n=0 は不可) (デフォルト値) ed = -1

n > 0 : データの先頭から n バイト目

n < 0 : データの末尾から n バイト目

※ n バイト目が全角文字の 2 バイト目の時はその全角文字が終点になります。

pc=[-]“string” : 文字列を始点・終点に指定

マイナス (-) を付けた場合その文字列を含める。st、ed の直後に記述する。

st=n, pc=“string” n バイト目以降にある “string” より後から

st=n, pc=-“string” n バイト目以降にある “string” から

ed=n, pc=“string” n バイト目以前にある “string” より前まで

ed=n, pc=-“string” n バイト目以前にある “string” まで

tc=“string” : 変換文字列の指定

ex1) データ全体を “?” に置き換える

```
rep, “?”
```

変換元 : Create!Form

変換後 : ?

ex2) データの先頭から 4 バイト目以降を “” に置き換える (削除する)

```
rep, st=4, “”
```

変換元 : wonderful

変換後 : won

ex3) データの末尾から 4 バイト目以前を “” に置き換える (= 削除する)

```
rep, ed=-4, “”
```

変換元 : wonderful

変換後 : ful

ex4) データの先頭から 3 バイト目以降にある “ 年 ” より後、末尾から 2 バイト目以前にある “ 日 ” から前を “ ” に置き換える (= 削除する)

rep, st=3, pc=“ 年 ”, ed=-2, pc=-“ 日 ”, “ ”

変換元 : 2002 年 7 月 1 日月曜日

変換後 : 2002 年月曜日

ex5) データ中にある “ ” (スペース) を “ ” に置き換える (= 削除する)

rep, tc=“ ”, “ ”

変換元 : 1234 567 89

変換後 : 123456789

ex6) データの先頭から 1 バイト目にある “-” を “ Δ ” に置き換える (“-” がなければ編集しない)

rep, st=1, tc=-“ ”, “ Δ ”

変換元 : -365

変換後 : Δ 365

ex7) データの先頭から最初の “ ” (スペース) 以降を削除する

rep, st=1, pc=-“ ”, “ ”

変換元 : 鈴木 紀子

変換後 : 鈴木

ex8) データの末尾から最初の “ ” (スペース) より前を削除する

rep, ed=-1, pc=-“ ”, “ ”

変換元 : 鈴木 紀子

変換後 : 紀子

#### 4-3. cnv コマンド

編集データの一部または全部をパラメータで指定した変換タイプに変換します。

小文字→大文字、大文字→小文字、半角→全角、全角→半角、ひらがな→カタカナ、カタカナ→ひらがな、数字→漢数字といった変換が可能です。

##### フォーマット文字列：

cnv < , st = [ - ] n [ , pc = [ - ] “string” ] / , ed = [ - ] n [ , pc = [ - ] “string” ] > , edit\_type

st=[-]n: 始点の指定 (デフォルト値) st = 1

n > 0: データの先頭から n バイト目

n < 0: データの末尾から n バイト目

※ n バイト目が全角文字の 2 バイト目の時はその全角文字が始点になります。

ed=[-]n: 終点の指定 (デフォルト値) ed = -1

n > 0: データの先頭から n バイト目

n < 0: データの末尾から n バイト目

※ n バイト目が全角文字の 2 バイト目の時はその全角文字が終点になります。

pc=[-]"string" : 文字列を始点・終点に指定  
 マイナス (-) を付けた場合その文字列を含める。st、ed の直後に記述する。  
 st=n, pc="string" n バイト目以降にある "string" より後から  
 st=n, pc=-"string" n バイト目以降にある "string" から  
 ed=n, pc="string" n バイト目以前にある "string" より前まで  
 ed=n, pc=-"string" n バイト目以前にある "string" まで

edit\_type : 変換タイプ

u : 半角英小文字を半角英大文字に変換する

l : 半角英大文字を半角英小文字に変換する

s : 全角文字を半角文字に変換する

d : 半角文字を全角文字に変換する

h : 全角カタカナをひらがなに変換する

k : ひらがなを全角カタカナに変換する

c : 半角数値データを漢字による金額表示に変換する

※ s、d、h、k の変換では変換対象になる文字のみ変換して、その他の文字は変換しません。

※ c の変換は 16 桁以内の半角整数値に対してのみ変換します。

ex1) データの先頭から 2 バイト目以降を小文字に変換する

cnv, st=2, l

変換元 : WONDERFUL!

変換後 : Wonderful!

ex2) データを全角文字に変換する

cnv, d

変換元 : Wonderful!

変換後 : W o n d e r f u l !

ex3) データ中のカタカナをひらがなに変換する

cnv, h

変換元 : ドラやき

変換後 : だらやき

ex4) 数値データを漢字表示にする

cnv, c

変換元 : 1234567890

変換後 : 壹拾貳億參千四百伍拾六萬七千八百九拾

#### 4-4. ded コマンド

編集データ（半角数値）の切り上げ・切り捨て・四捨五入・桁揃えを行います。

**フォーマット文字列：**

ded , fg = [ + | - | +- ] n

n : 小数点以下の桁数

+ : 切り上げ ( $0 \leq n \leq 11$ )

- : 切り捨て ( $0 \leq n \leq 11$ )

+- : 四捨五入 ( $0 \leq n \leq 11$ )

なし : 桁揃え ( $n \geq 0$ )

※ ded コマンドによる数値データの編集では整数部が 13 桁以下の必要があります。14 桁以上だと誤差が生じるため、正確な計算ができません。

ex1) 小数点以下 3 桁目を切り上げ

ded, fg=+2

変換元 : 123. 456

変換後 : 123. 46

ex2) 小数点以下 3 桁目を切り捨て

ded, fg=-2

変換元 : 123. 456

変換後 : 123. 45

ex3) 小数点以下 3 桁目を四捨五入

ded, fg=+-2

変換元 : 123. 456

変換後 : 123. 46

ex4) 小数点以下を 5 桁でそろえる

ded, fg=5

変換元 : 123. 456

変換後 : 123. 45600

ex5) 小数点以下を 2 桁でそろえる

ded, fg=2

変換元 : 123. 456

変換後 : 123. 45

#### 4-5. adj コマンド

パラメータで指定した幅で編集データの文字間隔を調整します。

**フォーマット文字列：**

adj , wd = n , regulate\_type

n : 表示幅 (バイト数) ( $n \geq$  [編集データの長さ])

regulate\_type : 調整タイプ

1 : 編集データが半角文字の場合は半角スペース、全角文字の場合は全角スペースで文字間隔を調整する

2: 編集データが全角文字の場合に全角スペース及び半角スペースで文字間隔を調整する  
 ※ adj コマンドでは、変換元が半角文字のみもしくは全角文字のみの必要があります。半角文字と全角文字が変換元に混在している場合、変換ができません。

ex1) 調整タイプ1 (編集データが半角文字)

adj, wd=24, 1  
 変換元: ABCDE  
 変換後: A B C D E

ex2) 調整タイプ1 (編集データが全角文字)

adj, wd=24, 1  
 変換元: A B C D E  
 変換後: A B C D E

ex3) 調整タイプ2

adj, wd=24, 2  
 変換元: A B C D E  
 変換後: A B C D E

#### 4-6. gen コマンド

パラメータで指定した文字列を、編集データ（半角数値）の値だけ繰り返し出力します。

**フォーマット文字列:**

gen, "string"

ex)

gen, "#"  
 変換元: 5  
 変換後: #####

#### 4-7. ext コマンド

パラメータで指定した dff ファイル名からデータ編集を利用します。

dff ファイルは事前にユーザー設定(共通)フォルダ直下の [dff] フォルダへ作成しておきます。  
 ユーザー設定(共通)フォルダはマネージャの [ヘルプ]-[バージョン情報]-[バージョン情報詳細] で確認できます。

ext コマンドを使用して dff ファイル名を指定することで、その dff ファイル内に記述されたフリーフォーマットコマンドを使用してデータ編集を行うことができます。

**dff ファイルフォーマット**

dff ファイルへフリーフォーマットコマンドを記述します。記述する際は、コマンドを括弧 ( ) で括る必要があります。

なお、複数のフリーフォーマットコマンドを組み合わせで記述することもできます。その場合は 1 行ごとに記述し、各コマンドを括弧 ( ) で括ります。

```
ex)
(rep, st=2, "")
(rep, ed=-1, tc="4", "平成")
(rep, ed=-1, tc="3", "昭和")
(rep, ed=-1, tc="2", "大正")
(rep, ed=-1, tc="1", "明治")
.
.
.
```

#### << 注意 >>

dff ファイルの文字コードは SJIS で記述する必要があります。

また、フリーフォーマット文字列の上限は 3072 バイトです。上限を超えて記述することはできません。

#### フォーマット文字列：

```
ext, "dff ファイル名"
```

※ dff ファイル名は半角英数字で拡張子を含めて指定する必要があります。

```
ex)
ext, "sample. dff"
```

## 4-8. コマンドの複数指定

1 回のデータ編集においてコマンドは複数指定することができます。その場合は各コマンドを括弧 ( ) で括ります。

(コマンド, パラメータ [, パラメータ, …]) (コマンド, パラメータ [, パラメータ, …]) …

```
ex1) (ins, st=4, "年") (ins, st=-2, "月") (ins, st=-0, "日")
```

変換元：20020701

変換後：2002 年 07 月 01 日

```
ex2) (adj, wd=14, 1) (cnv, ed=1, u) (cnv, st=8, ed=8, u) (ins, st=0, "***") (ins, st=-0, "***")
```

変換元：create!form

変換後：\*\*\*Create!Form\*\*\*

#### << 注意 >>

データ編集作成ダイアログ上でのフリーフォーマット文字列の上限は 1024 バイトです。

また、dff ファイル上でのフリーフォーマット文字列の上限は 3072 バイトです。上限を超えて記述することはできません。

# Create!Form 11

データ編集 第3版

---

|     |                            |
|-----|----------------------------|
| 発行日 | 2019年4月                    |
| 発行者 | インフォテック株式会社                |
|     | 〒160-0023 東京都新宿区西新宿 7-5-25 |