

# **機能仕様書 出力データ抽出**

富士通株式会社  
2010 年 2 月 15 日

## 更新履歴

日付	版数	担当	備考
2009/8/7	0.1 版	FST)	
2010/1/12	1.0 版	FST)	関数追加
2010/2/15	1.1 版	FST)	コマンド名変更、コマンドエイリアス追加

まえがき

本書は、出力データ抽出に求められる機能要件まとめたものです。  
また、本書はプロト版の為、機能・性能改善するのに当たり、予告なしに変更する場合があります。

- 未実装、仕様未確定  
機能が未実装であったり、仕様が未確定の部分は、本文中で網駆け（機能）表記しています。  
実装や性能改善するのに当たり、予告なしに変更する場合があります。

- 表記上規則

記号	意味
{ <u>ABC</u>   EFG}	{ }内の文字列の 1 つを選択することを示します。省略した場合、“_”（アンダーライン）の文字列が選択されたことを示します。
[ABC]	[ ]で囲まれた文字列は省略できることを示します。

## 目次

第1章 機能要件.....	1
1.1. 利用要件.....	2
第2章 機能.....	3
2.1. 定型抽出.....	3
2.1.1. 行抽出.....	4
2.1.1.1. 行番号による行抽出.....	4
2.1.1.2. 正規表現による行抽出.....	4
2.1.2. 列抽出.....	5
2.1.2.1. 列番号による列抽出.....	5
2.1.2.2. 正規表現による列抽出.....	5
2.1.3. ブロック抽出.....	6
2.1.3.1. 行番号(行)と列番号(列)によるブロック抽出.....	6
2.1.3.2. 行番号(行)と正規表現(列)によるブロック抽出.....	6
2.1.3.3. 正規表現(行)と列番号(列)によるブロック抽出.....	6
2.1.3.4. 正規表現(行)と正規表現(列)によるブロック抽出.....	7
2.2. ユーザ抽出.....	8
2.3. 抽出対象データ.....	8
2.3.1. 列の定義.....	8
2.3.2. 行の定義.....	8
2.4. 抽出結果.....	9
第3章 抽出コマンド.....	10
3.1. 抽出対象データ定義コマンド.....	11
3.1.1. 書式.....	11
3.1.2. パラメタ.....	11
3.1.3. 復帰値.....	11
3.1.4. 記述例.....	11
3.2. 抽出条件定義コマンド.....	12
3.2.1. condition.....	13
3.2.1.1. 書式.....	13
3.2.1.2. パラメタ.....	13
3.2.1.3. 記述例.....	15
3.2.2. extract_line_nn/extract_line.....	17
3.2.2.1. 書式.....	17
3.2.2.2. パラメタ.....	17
3.2.2.3. 記述例.....	18
3.2.3. extract_line_nr.....	19
3.2.4. extract_line_rn/extract_line_r.....	20
3.2.4.1. 書式.....	20
3.2.4.2. パラメタ.....	20
3.2.4.3. 記述例.....	21
3.2.5. extract_line_rr.....	22
3.2.5.1. 書式.....	22
3.2.5.2. パラメタ.....	22
3.2.5.3. 記述例.....	22
3.2.6. extract_column_nn/extract_column.....	23
3.2.6.1. 書式.....	23
3.2.6.2. パラメタ.....	23

3.2.6.3. 記述例.....	24
3.2.7. extract_column_nr.....	25
3.2.8. extract_column_rn/extract_column_r.....	26
3.2.8.1. 書式.....	26
3.2.8.2. パラメタ .....	26
3.2.8.3. 記述例.....	27
3.2.9. extract_column_rr.....	28
3.2.9.1. 書式.....	28
3.2.9.2. パラメタ .....	28
3.2.9.3. 記述例.....	28
3.2.10. remove_line_nn/remove_line.....	29
3.2.10.1. 書式.....	29
3.2.10.2. パラメタ .....	29
3.2.10.3. 記述例.....	29
3.2.11. remove_line_nr.....	30
3.2.12. remove_line_rn/remove_line_r.....	31
3.2.12.1. 書式.....	31
3.2.12.2. パラメタ .....	31
3.2.12.3. 記述例.....	31
3.2.13. remove_line_rr.....	32
3.2.13.1. 書式.....	32
3.2.13.2. パラメタ .....	32
3.2.13.3. 記述例.....	32
3.2.14. remove_column_nn/remove_column.....	33
3.2.14.1. 書式.....	33
3.2.14.2. パラメタ .....	33
3.2.14.3. 記述例.....	33
3.2.15. remove_column_nr.....	34
3.2.16. remove_column_rn/remove_column_r.....	35
3.2.16.1. 書式.....	35
3.2.16.2. パラメタ .....	35
3.2.16.3. 記述例.....	35
3.2.17. remove_column_rr.....	36
3.2.17.1. 書式.....	36
3.2.17.2. パラメタ .....	36
3.2.17.3. 記述例.....	36
3.3. 抽出指示コマンド.....	37
3.3.1. 書式.....	37
3.3.2. パラメタ .....	37
3.3.3. 復帰値.....	37
第4章 ユーザ関数.....	38
4.3.1. ユーザ関数の記述場所 .....	38
4.3.2. ユーザ関数の呼出し.....	39
4.3.3. 抽出対象ファイルのデータの読込み・書込み等 .....	39
4.3.4. ユーザ関数が返却する抽出結果.....	40
第5章 ユーザ関数内で使える関数.....	41
5.1. get_line_num.....	41
5.1.1. 書式.....	41

5.1.2. パラメタ .....	41
5.1.3. 記述例 .....	41
5.2. seek_line.....	42
5.2.1. 書式.....	42
5.2.2. パラメタ .....	42
5.2.3. 記述例 .....	42
5.3. get_line.....	43
5.3.1. 書式.....	43
5.3.2. パラメタ .....	43
5.3.3. 記述例 .....	43
5.4. add_data.....	44
5.4.1. 書式.....	44
5.4.2. パラメタ .....	44
5.4.3. 記述例 .....	44
5.5. del_data.....	45
5.5.1. 書式.....	45
5.5.2. パラメタ .....	45
5.5.3. 記述例 .....	45

## 第1章 機能要件

アプリケーションが出力したデータに対してフィルタリングを複数回行い、必要情報のみを簡単に抽出したい。

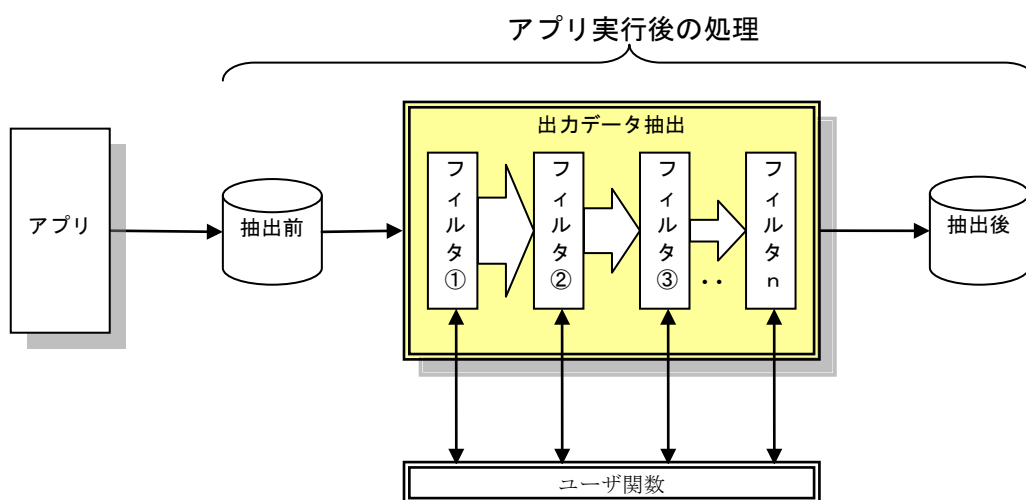
データ抽出(フィルタリング)する方法としては、以下が考えられる。

- あるルールに従って必要情報を抽出する
- 条件式などをプログラミングして必要情報を抽出する

プログラミングを要するデータの抽出方法は、ツールによる仕組みという分類から外れているため、ユーザが実際にプログラミングする部分については本検討から除外する。

よって、以下について検討する。

- あるルールに従って必要情報を抽出する  
→ 定型抽出
- 条件式などをプログラミングして必要情報を抽出する為のユーザインターフェース  
→ ユーザ抽出



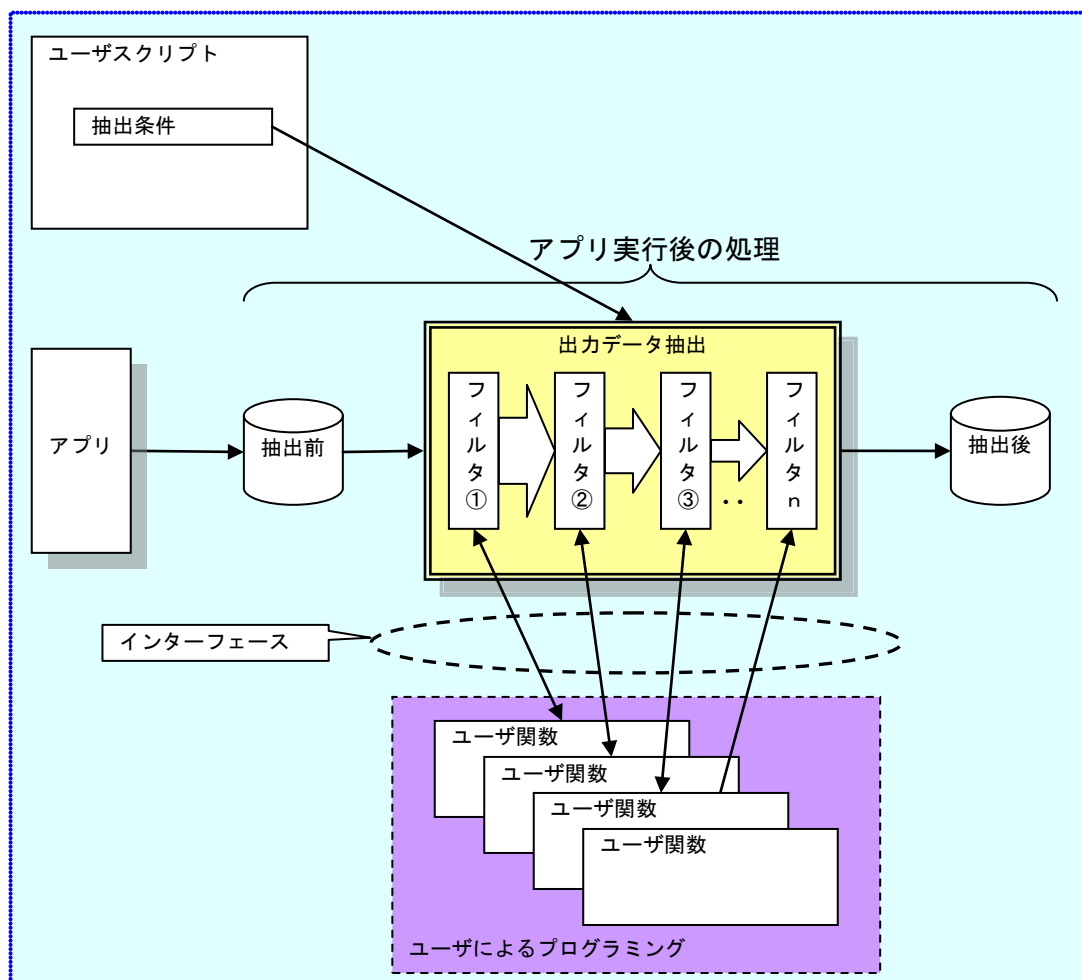
## 1.1. 利用要件

アプリケーションは、デバック情報など結果の評価には不必要な情報も含んだ形でデータ出力する場合がある。その場合、必要情報の抽出が必要となるが、時間が掛かるしミスなどの危険性がある為、手作業はしたくない。

抽出条件を与えるだけで、データ抽出が出来るくらいの簡単さを求めている。また、型にはまる抽出だけでなく、複雑な抽出条件などをユーザ自ら追加できるような仕組みを求めている。

抽出条件が重要なポイントとなる。

利用イメージは次のようになる。





## 第2章 機能

第1章で記述した要件を満たすために、実現する機能について述べる。

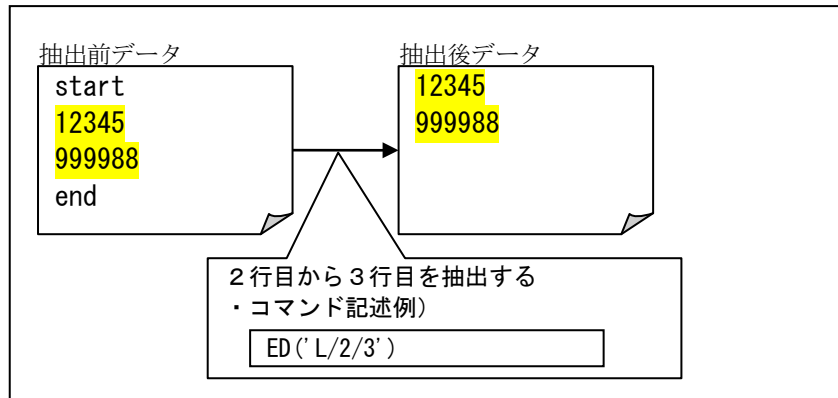
### 2.1. 定型抽出

データから一部を抽出するためには、抽出するための抽出条件が必要となる。抽出は、以下の5パターンを想定し実現する。

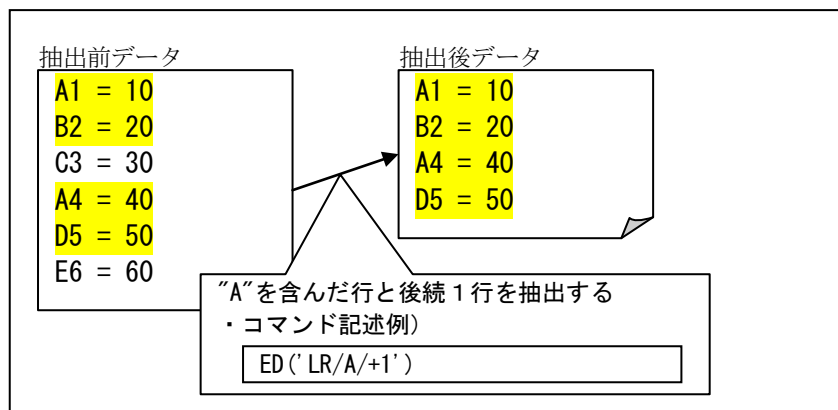
- 行抽出
  - ・行番号による行抽出
  - ・正規表現による行抽出
- 列抽出
  - ・列番号による列抽出
  - ・正規表現による列抽出
- ブロック抽出
  - ・「行抽出」と「列抽出」を組み合わせた抽出

## 2.1.1. 行抽出

### 2.1.1.1. 行番号による行抽出

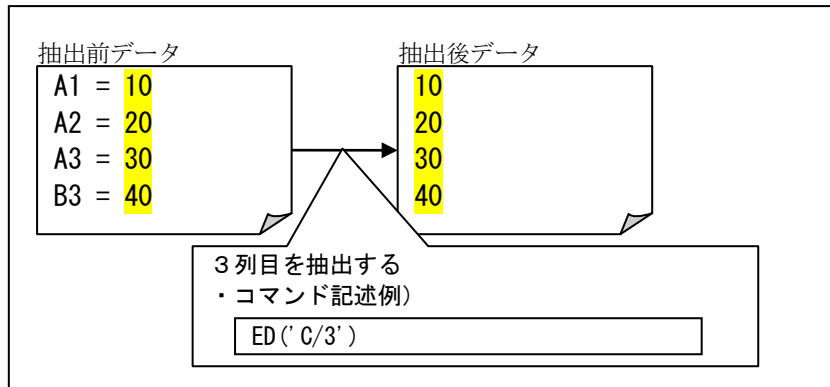


### 2.1.1.2. 正規表現による行抽出

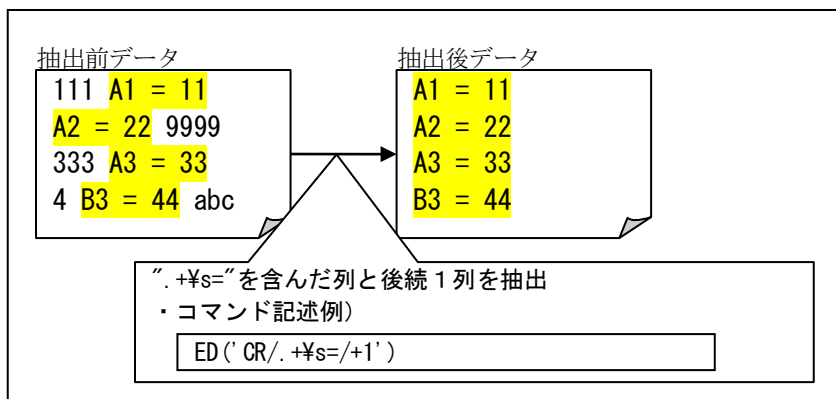


## 2.1.2. 列抽出

### 2.1.2.1. 列番号による列抽出

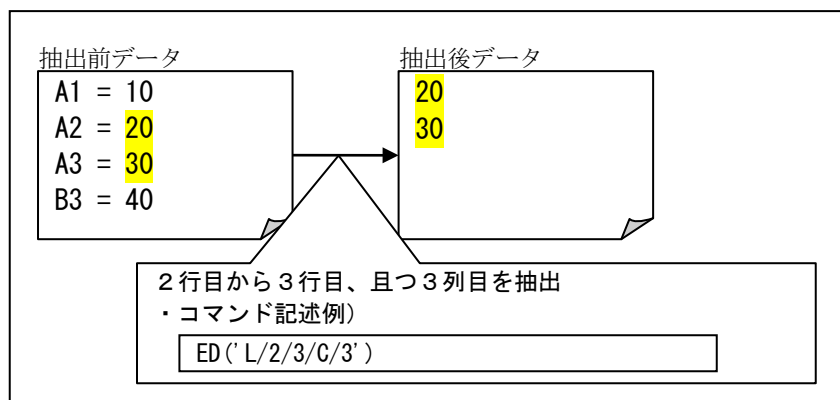


### 2.1.2.2. 正規表現による列抽出

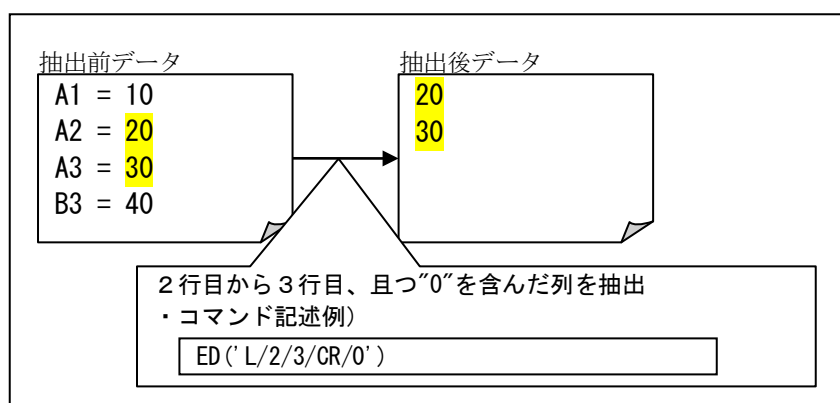


### 2.1.3. ブロック抽出

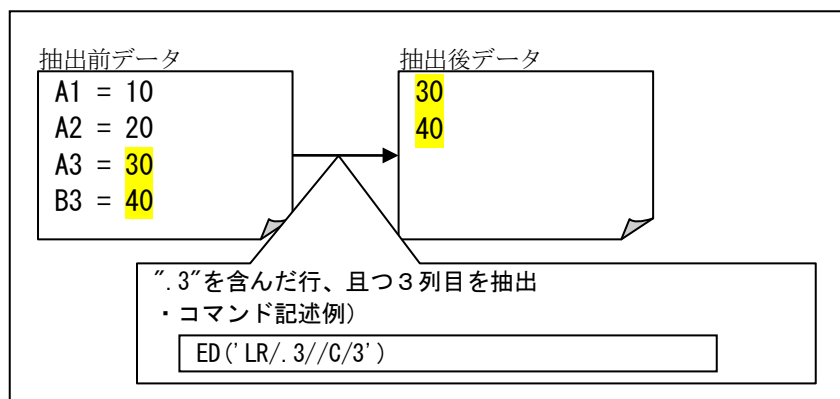
#### 2.1.3.1. 行番号(行)と列番号(列)によるブロック抽出



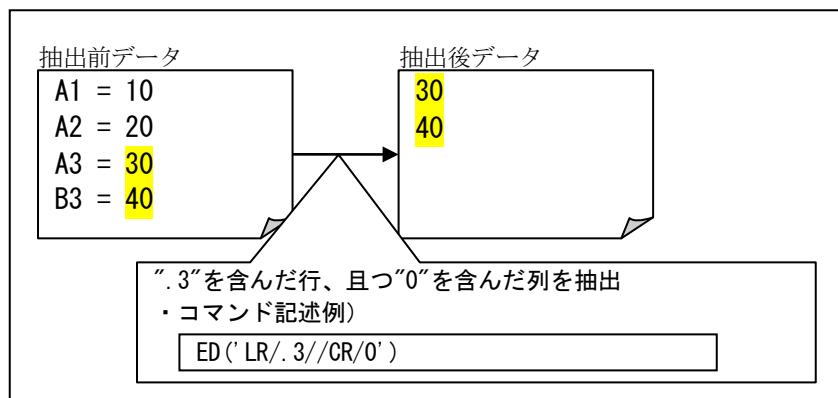
#### 2.1.3.2. 行番号(行)と正規表現(列)によるブロック抽出



#### 2.1.3.3. 正規表現(行)と列番号(列)によるブロック抽出



#### 2.1.3.4. 正規表現(行)と正規表現(列)によるブロック抽出



## 2.2. ユーザ抽出

複雑な抽出条件などをユーザ自ら追加できるように、ユーザ関数とのインターフェース仕様を定める。詳細は、「第4章 ユーザ関数」を参照のこと。

## 2.3. 抽出対象データ

初回抽出の場合、テキスト形式のデータを格納したファイルまたは変数を対象データとする。2回目以降の抽出の場合、直前に抽出した結果を対象データとする。

### 2.3.1. 列の定義

空白・タブ・改行・,(カンマ)を区切り文字とし、その区切り文字で分割した文字列を列とする。区切り文字が連続して続いている場合には、一つの区切り文字として扱う。

尚、単純に区切り文字で分割したものを列と定義している為、前後を' (シングルクォート) や" (ダブルクォート) で囲った文字列内に区切り文字があるデータの取り扱いには推奨しない。

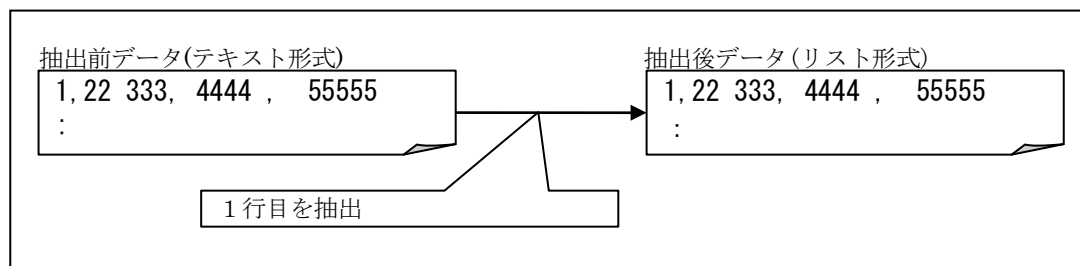
### 2.3.2. 行の定義

改行を区切り文字とし、改行までのデータを1行とする。

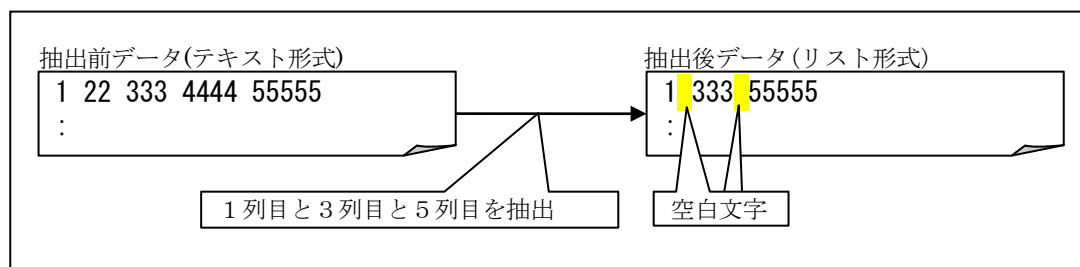
## 2.4. 抽出結果

抽出結果は、本ツールの復帰値(リスト形式)として返す。

行を抽出した場合、抽出対象データをそのままの形式で出力する。



複数列を抽出した場合、列と列の間に空白文字(区切り文字)を付与して出力する。



## 第3章 抽出コマンド

データ抽出の定義と指示は、抽出コマンドを用いて記述する。抽出コマンドは、スクリプトのアプリ実行後処理内に明記する。

＜抽出定義＞

- 抽出対象データ定義コマンド
- 抽出条件定義コマンド

<抽出指示>

- ## ● 抽出指示コマンド

抽出コマンドの記述順は、以下の通り。

1. 抽出対象データ定義コマンド
  2. 抽出条件定義コマンド
  - : :
  - : 抽出条件定義コマンド
  3. 抽出指示コマンド
- } (複数記述可)

## 記述例)

：(前略)

## #＜アプリ実行後処理＞

```
my $After = data_extractor->new('file:入力ファイル名');
```

```
$After->condition('抽出条件');
```

```
$After->condition('抽出条件');
```

1

```
$After->condition('抽出条件');
```

```
$After->execute();
```

(後略)

$$\vdots$$

## 抽出対象データ定義コマンド

## 抽出条件定義コマンド

## 抽出指示コマンド



### 3.1. 抽出対象データ定義コマンド

抽出対象とするデータの定義を行う。

#### 3.1.1. 書式

```
$オブジェクト名 = data_extractor->new("抽出対象データ", "ユーザ指定最大バッファ数");
```

#### 3.1.2. パラメタ

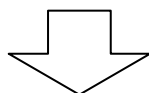
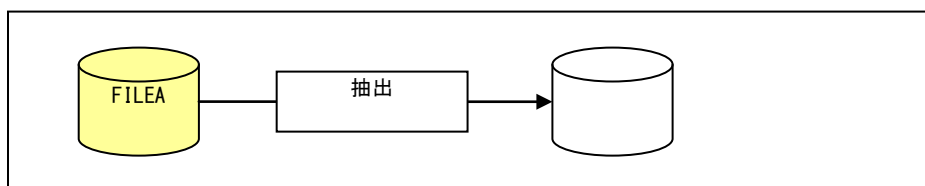
- オブジェクト名  
データ抽出するためのオブジェクトを格納する変数名。
- 抽出対象データ  
抽出対象とするデータを格納した変数名。  
または、  
抽出対象とするデータを格納したファイル名（絶対パス付き）
- ユーザ指定最大バッファ数  
ユーザ関数内で使用するバッファ行数。  
ユーザは、この指定を超えた範囲の行参照は出来ない。

#### 3.1.3. 復帰値

なし。

#### 3.1.4. 記述例

/home/pjo/FILEA を抽出対象とする。



```
$オブジェクト名 = EF('file:/home/pjo/FILEA');
```

### 3.2. 抽出条件定義コマンド

抽出を行う条件の定義を行う。抽出目的に合わせた定義コマンドを用いる。

複合条件抽出

- condition

行抽出

- extract\_line\_nn/extract\_line
- extract\_line\_nr
- extract\_line\_rn/extract\_line\_r
- extract\_line\_rr

列抽出

- extract\_column\_nn/extract\_column
- extract\_column\_nr
- extract\_column\_rn/extract\_column\_r
- extract\_column\_rr

否定行抽出

- remove\_line\_nn/remove\_line
- remove\_line\_nr
- remove\_line\_rn/remove\_line\_r
- remove\_line\_rr

否定列抽出

- remove\_column\_nn/remove\_column
- remove\_column\_nr
- remove\_column\_rn/remove\_column\_r
- remove\_column\_rr

### 3.2.1. condition

抽出条件の定義を行う。

#### 3.2.1.1. 書式

\$オブジェクト名->condition(抽出条件[, 抽出条件[, ...]]);

抽出条件 = { '定型抽出' | ユーザー抽出 }

※定型抽出 = 行番号や列番号、正規表現を用いた行・列・ブロック抽出

※ユーザー抽出 = ユーザー関数によるユーザー抽出

定型抽出

行抽出

行番号指定 : [!]L/{行番号 | end[-数字]}/[範囲指定]

正規表現指定 : [!]LR/比較値/[範囲指定]

列抽出

列番号指定 : [!]C/{列番号 | end[-数字]}/[範囲指定]

正規表現指定 : [!]CR/比較値/[範囲指定]

ブロック抽出 : 行抽出/列抽出

ユーザー抽出 : ['パッケージ::サブルーチン'[, 'ユーザー引数'[, ...]]]  
※大外の [ ] は、配列定義を指す

#### 3.2.1.2. パラメタ

➤ !

条件に該当する情報以外を抽出。

➤ 行番号

抽出対象データの先頭行を 1 としてカウントした行番号。

➤ 列番号

抽出対象データの先頭列を 1 としてカウントした列番号。

➤ 比較値

抽出する条件（正規表現）。

- end[-数字]
  - ・ end
    - 行抽出の場合、最終行。
    - 列抽出の場合、最終列。
  - ・ -数字
    - 行抽出の場合、最終行から数えた行数。
    - 列抽出の場合、最終列から数えた列数。

- 範囲指定
  - 抽出範囲の条件。
  - 抽出条件によって、記述できる指定方法が異なる。

抽出条件		範囲指定に記述できる指定方法					
		行番号	比較値	列番号	+範囲	-範囲	end[-数字]
行抽出	行番号指定	○	—	—	○	○	○
	正規表現指定	—	○	—	○	○	—
列抽出	列番号指定	—	—	○	○	○	○
	正規表現指定	—	○	—	○	○	—

※正規表現指定、且つ範囲指定を比較値で指定する場合の注意事項

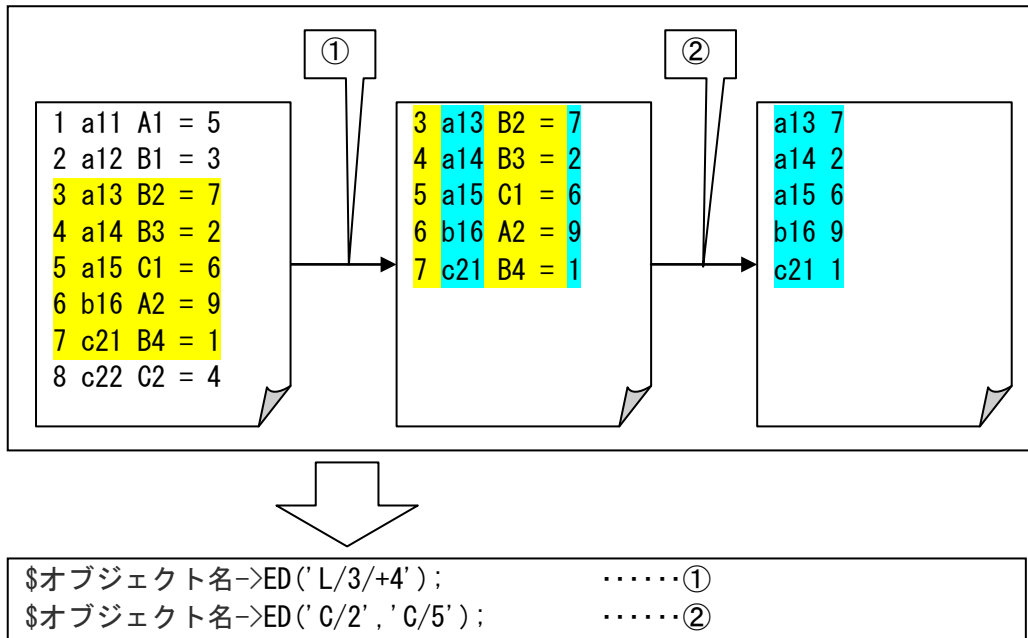
- ・ ' - ' (マイナス) に続いて数字を記述するには ' ¥ - ' と表記する。

- +範囲
  - 行抽出の場合、抽出対象範囲としたい後続行数。
  - 列抽出の場合、抽出対象範囲としたい後続列数。
  - +1、+2 のように数字の前に '+' を付けて明記。
- -範囲
  - 行抽出の場合、抽出対象範囲としたい先行行数。
  - 列抽出の場合、抽出対象範囲としたい先行列数。
  - 1、-2 のように数字の前に '-' を付けて明記。
- パッケージ
  - ユーザ関数(サブルーチン)を格納したパッケージ名。
  - ユーザ関数をユーザスクリプトに記述した場合、'main' を指定。
- サブルーチン
  - ユーザ関数名。
- ユーザ引数
  - ユーザ関数に渡す引数。

### 3.2.1.3. 記述例

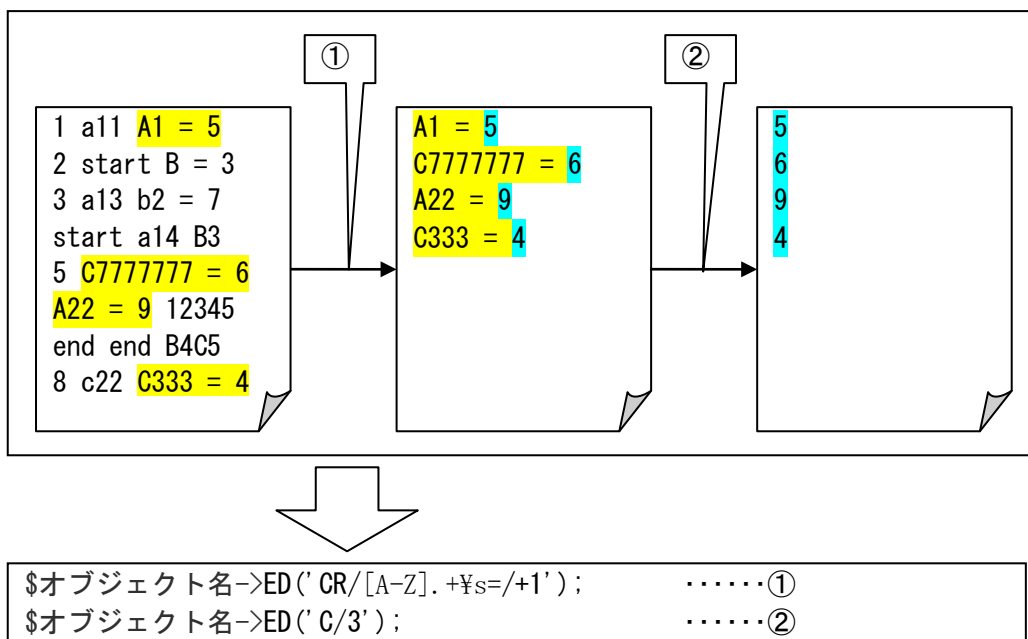
例 1)

- ① 3行目から後続4行を抽出する。
- ② 2列目と5列目を抽出する。



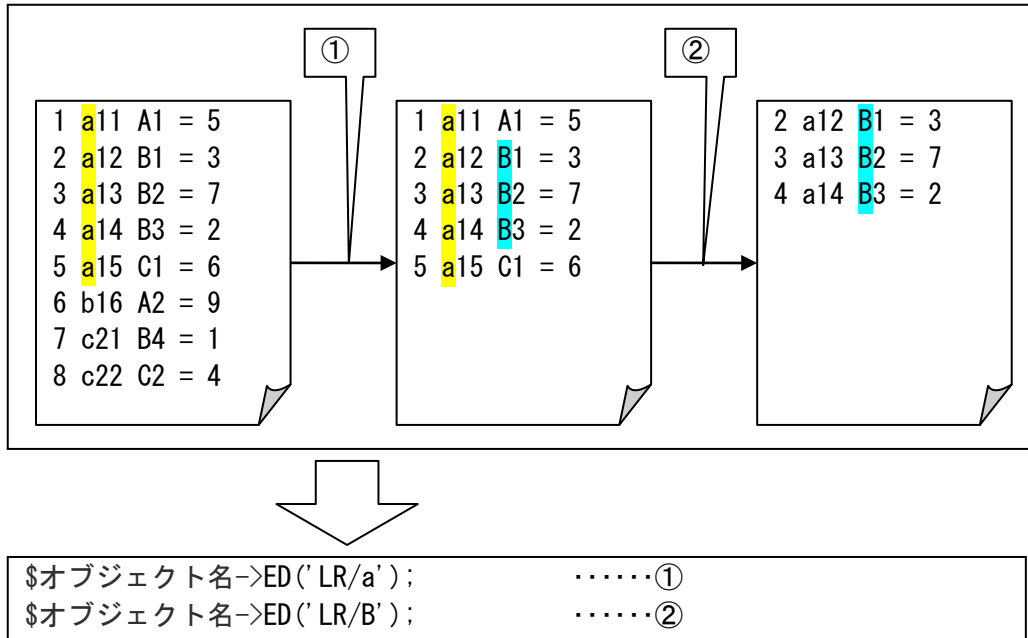
例 2)

- ① '[A-Z]. +¥s='を含む列から後続1列を抽出する。
- ② 3列目を抽出する。



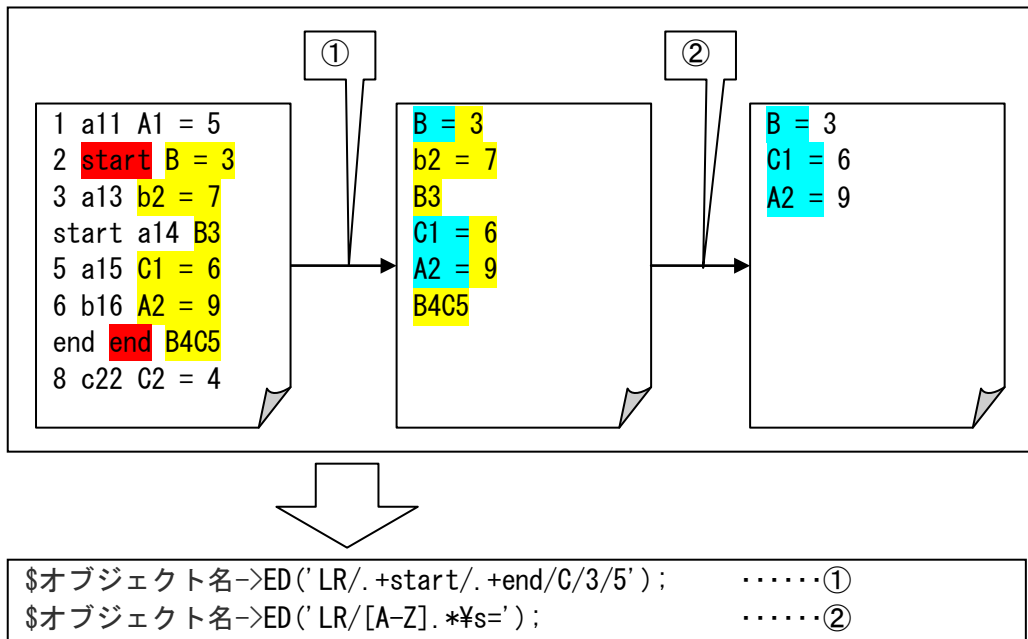
例 3)

- ① 'a' を含む行を抽出する。  
 ② 'B' を含む行を抽出する。



例 4)

- ① '.+start' から '.+end' までの間の行の内、3列目から5列目を抽出する。  
 ② '[A-Z].\*¥s=' を含む行を抽出する。



### 3.2.2. extract\_line\_nn/extract\_line

行番号指定による行抽出の定義を行う。

#### 3.2.2.1. 書式

\$オブジェクト名->extract\_line\_nn(抽出条件[, 抽出条件[, ...]]);

抽出条件 = {{行番号 | end[-数字]} | [{行番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

\$オブジェクト名->extract\_line(抽出条件[, 抽出条件[, ...]]);

抽出条件 = {{行番号 | end[-数字]} | [{行番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

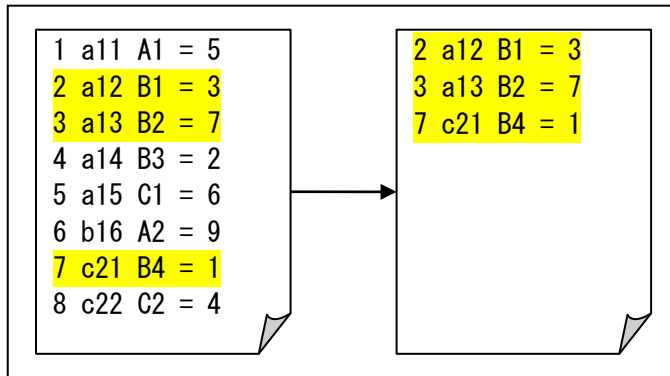
範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

#### 3.2.2.2. パラメタ

- 行番号  
抽出対象データの先頭行を 1 としてカウントした行番号。
- +範囲  
抽出対象範囲としたい後続行数。  
+1、+2 のように数字の前に '+' を付けて明記。
- -範囲  
抽出対象範囲としたい先行行数。  
-1、-2 のように数字の前に '-' を付けて明記。
- end[-数字]
  - end  
最終行。
  - -数字  
最終行から数えた行数。

### 3.2.2.3. 記述例

2行目から後続1行と最終行の先行1行の範囲を抽出する。



\$オブジェクト名->extract\_line\_nn([2, '+1'], 'end-1');



### 3.2.3. extract\_line\_nr

未実装

### 3.2.4. extract\_line\_rn/extract\_line\_r

正規表現 (範囲指定は相対行番号) 指定による行抽出の定義を行う。

#### 3.2.4.1. 書式

\$オブジェクト名->extract\_line\_rn(抽出条件[, 抽出条件[, ...]]);

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

\$オブジェクト名->extract\_line\_r(抽出条件[, 抽出条件[, ...]]);

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

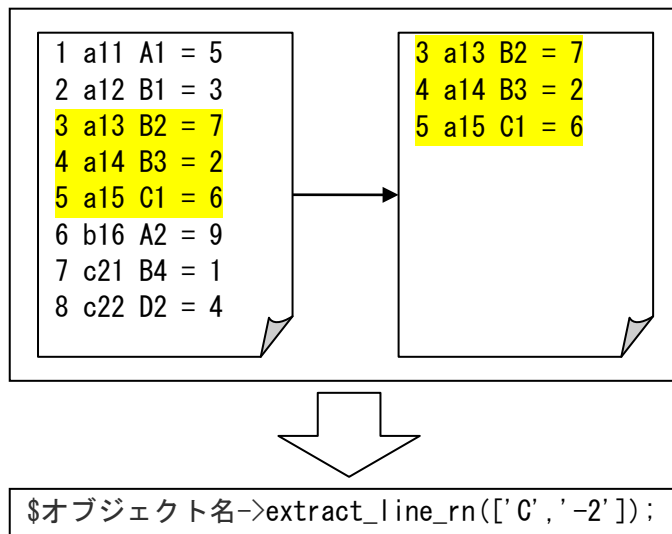
範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

#### 3.2.4.2. パラメタ

- 比較値  
抽出する条件 (正規表現)。
- 行番号  
抽出対象データの先頭行を 1 としてカウントした行番号。
- +範囲  
抽出対象範囲としたい後続行数。  
'1' と '+1' は同様に処理する。
- -範囲  
抽出対象範囲としたい先行行数。  
-1、-2 のように数字の前に '-' を付けて明記。
- end[-数字]
  - end  
最終行。
  - -数字  
最終行から数えた行数。

### 3.2.4.3. 記述例

'C'を含んだ行から先行2行までの範囲を抽出する。



### 3.2.5. extract\_line\_rr

正規表現指定による行抽出の定義を行う。

#### 3.2.5.1. 書式

```
$オブジェクト名->extract_line_rr(抽出条件[, 抽出条件[, ...]]);
```

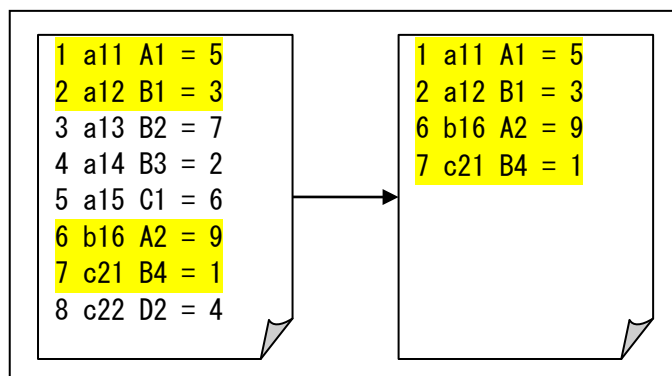
抽出条件 = {比較値 | [比較値, 比較値] }  
※ [ ] は、配列定義を指す

#### 3.2.5.2. パラメタ

- 比較値  
抽出する条件（正規表現）。

#### 3.2.5.3. 記述例

'A'を含んだ行から'B'を含んだ範囲を抽出する。



```
$オブジェクト名->extract_line_rr(['A', 'B']);
```

### 3.2.6. extract\_column\_nn/extract\_column

行番号指定による列抽出の定義を行う。

#### 3.2.6.1. 書式

```
$オブジェクト名->extract_column_nn(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {[列番号 | end[-数字]] | [{列番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

範囲指定 = {列番号 | end[-数字] | +範囲 | -範囲}

```
$オブジェクト名->extract_column(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {[列番号 | end[-数字]] | [{列番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

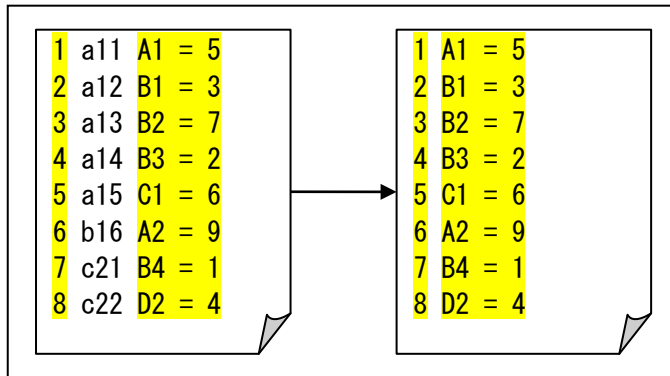
範囲指定 = {列番号 | end[-数字] | +範囲 | -範囲}

#### 3.2.6.2. パラメタ

- 列番号  
抽出対象データの先頭列を 1 としてカウントした列番号。
- +範囲  
抽出対象範囲としたい後続列数。  
+1、+2 のように数字の前に '+' を付けて明記。
- -範囲  
抽出対象範囲としたい先行列数。  
-1、-2 のように数字の前に '-' を付けて明記。
- end[-数字]
  - end  
最終列。
  - -数字  
最終列から数えた列数。

### 3.2.6.3. 記述例

1 列目と最終列の 2 列前から最終列の範囲を抽出する。



```
$オブジェクト名->extract_column_nn(1, ['end-2', 'end']);
```

### 3.2.7. extract\_column\_nr

未実装

### 3.2.8. extract\_column\_rn/extract\_column\_r

正規表現 (範囲指定は相対列番号) 指定による列抽出の定義を行う。

#### 3.2.8.1. 書式

\$オブジェクト名->extract\_column\_rn(抽出条件[, 抽出条件[, ...]]);

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

範囲指定 = {列号 | end[-数字] | +範囲 | -範囲}

\$オブジェクト名->extract\_column\_r(抽出条件[, 抽出条件[, ...]]);

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

範囲指定 = {列号 | end[-数字] | +範囲 | -範囲}

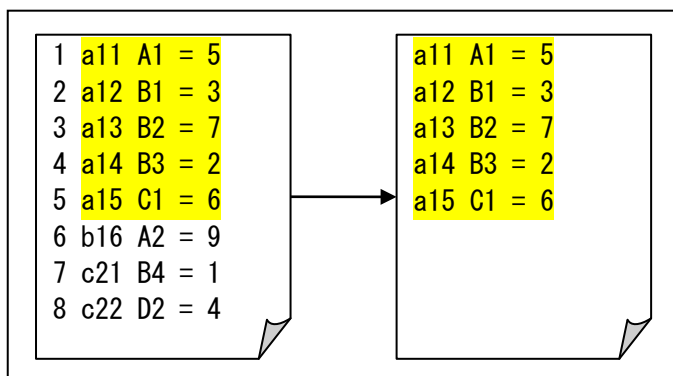
#### 3.2.8.2. パラメタ

- 比較値  
抽出する条件 (正規表現)。
- 列番号  
抽出対象データの先頭列を 1 としてカウントした列番号。
- +範囲  
抽出対象範囲としたい後続列数。  
'1' と '+1' は同様に処理する。
- -範囲  
抽出対象範囲としたい先行列数。  
-1、-2 のように数字の前に '-' を付けて明記。
- end[-数字]
  - end  
最終列。
  - -数字  
最終列から数えた列数。



### 3.2.8.3. 記述例

'a'を含んだ列から最終列の範囲を抽出する。



\$オブジェクト名->extract\_column\_rn([' a ', ' end' ] );

### 3.2.9. extract\_column\_rr

正規表現指定による列抽出の定義を行う。

#### 3.2.9.1. 書式

```
$オブジェクト名->extract_column_rr(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {比較値 | [比較値, 比較値]}

※ [ ] は、配列定義を指す

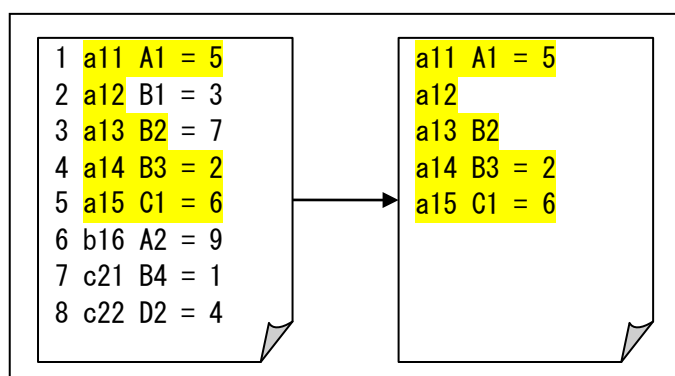
#### 3.2.9.2. パラメタ

##### ➤ 比較値

抽出する条件（正規表現）。

#### 3.2.9.3. 記述例

'a' を含んだ列から '2' を含んだ列の範囲を抽出する。



```
$オブジェクト名->extract_column_rr(['a','2']);
```

### 3.2.10. remove\_line\_nn/remove\_line

行番号指定による否定行抽出の定義を行う。

#### 3.2.10.1. 書式

```
$オブジェクト名->remove_line_nn(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {{行番号 | end[-数字]} | [{行番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

```
$オブジェクト名->remove_line(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {{行番号 | end[-数字]} | [{行番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

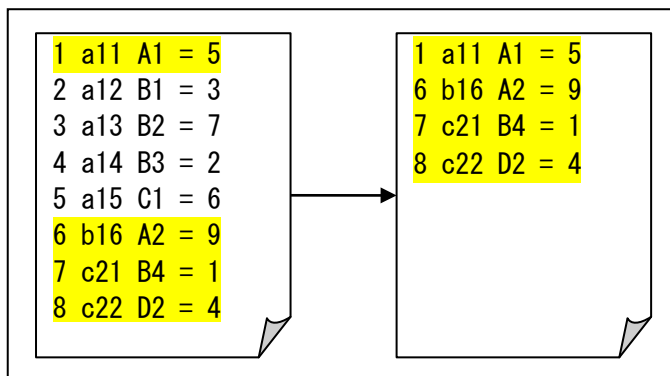
範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

#### 3.2.10.2. パラメタ

「3.2.2 extract\_line\_nn コマンド」を参照のこと。

#### 3.2.10.3. 記述例

2行目から後続3行以外の範囲を抽出する。



```
$オブジェクト名->remove_line_nn([2, '+1']);
```

### 3.2.11. remove\_line\_nr

未実装

### 3.2.12. remove\_line\_rn/remove\_line\_r

正規表現 (範囲指定は相対行番号) 指定による行抽出の定義を行う。

#### 3.2.12.1. 書式

```
$オブジェクト名->remove_line_rn(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

```
$オブジェクト名->remove_line_r(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

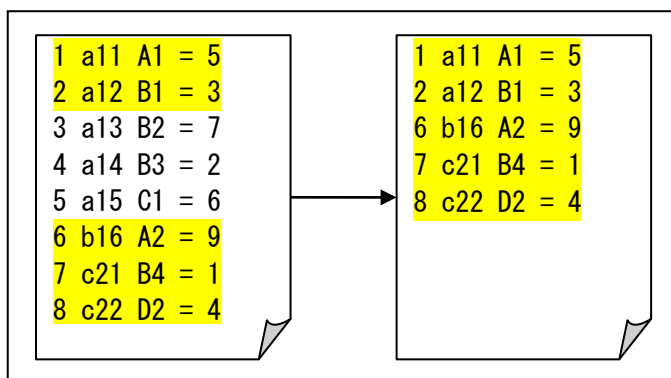
範囲指定 = {行番号 | end[-数字] | +範囲 | -範囲}

#### 3.2.12.2. パラメタ

「3.2.4 extract\_line\_rn コマンド」を参照のこと。

#### 3.2.12.3. 記述例

'C'を含んだ行から先行2行以外の範囲を抽出する。



```
$オブジェクト名->remove_line_rn(['C', '-2']);
```

### 3.2.13. remove\_line\_rr

正規表現指定による行抽出の定義を行う。

#### 3.2.13.1. 書式

```
$オブジェクト名->remove_line_rr(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {比較値 | [比較値, 比較値]}

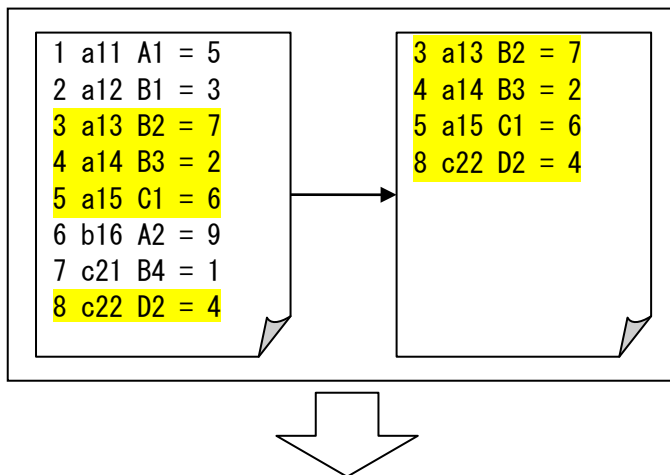
※ [ ] は、配列定義を指す

#### 3.2.13.2. パラメタ

「3.2.5 extract\_line\_rr コマンド」を参照のこと。

#### 3.2.13.3. 記述例

‘A’を含んだ行から‘B’を含んだ行以外の範囲を抽出する。



```
$オブジェクト名->remove_line_rr(['A', 'B']);
```

### 3.2.14. remove\_column\_nn/remove\_column

行番号指定による否定列抽出の定義を行う。

#### 3.2.14.1. 書式

```
$オブジェクト名->remove_column_nn(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {{列番号 | end[-数字]} | [{列番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

範囲指定 = {列番号 | end[-数字] | +範囲 | -範囲}

```
$オブジェクト名->remove_column(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {{列番号 | end[-数字]} | [{列番号|end[-数字]}, 範囲指定] }

※ [ ] は、配列定義を指す

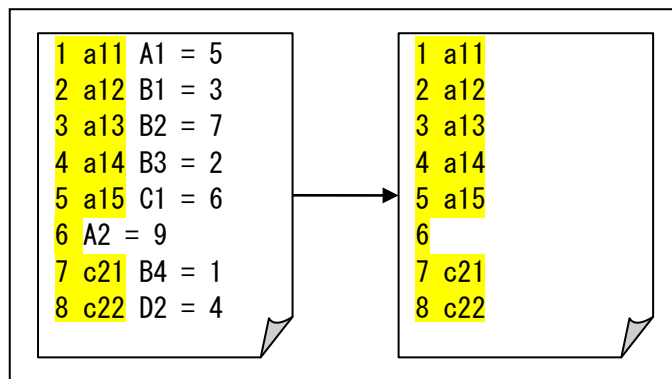
範囲指定 = {列番号 | end[-数字] | +範囲 | -範囲}

#### 3.2.14.2. パラメタ

「3.2.6 extract\_column\_nn コマンド」を参照のこと。

#### 3.2.14.3. 記述例

最終列から先行2列以外の範囲を抽出する。



```
$オブジェクト名->remove_column_nn(['end', '-2']);
```

### 3.2.15. remove\_column\_nr

未実装



### 3.2.16. remove\_column\_rn/remove\_column\_r

正規表現 (範囲指定は相対列番号) 指定による否定列抽出の定義を行う。

#### 3.2.16.1. 書式

```
$オブジェクト名->remove_column_rn(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

範囲指定 = {列号 | end[-数字] | +範囲 | -範囲}

```
$オブジェクト名->remove_column_r(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {比較値 | [比較値, 範囲指定]}

※ [ ] は、配列定義を指す

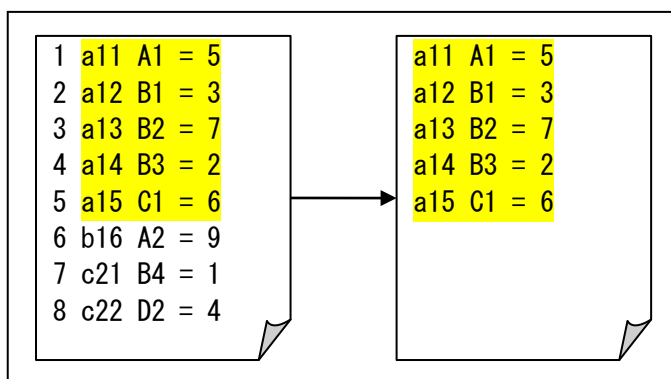
範囲指定 = {列号 | end[-数字] | +範囲 | -範囲}

#### 3.2.16.2. パラメタ

「3.2.8 extract\_column\_rn コマンド」を参照のこと。

#### 3.2.16.3. 記述例

'a'を含んだ列から最終列の範囲を抽出する。



```
$オブジェクト名->remove_column_rn(['a', 'end']);
```

### 3.2.17. remove\_column\_rr

正規表現指定による否定列抽出の定義を行う。

#### 3.2.17.1. 書式

```
$オブジェクト名->remove_column_rr(抽出条件[, 抽出条件[, ...]]);
```

抽出条件 = {比較値 | [比較値, 比較値]}

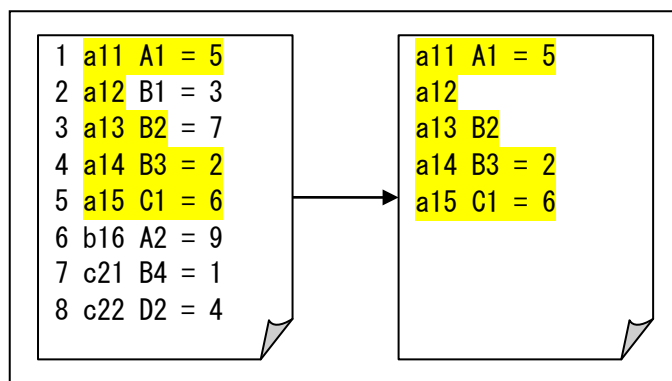
※ [ ] は、配列定義を指す

#### 3.2.17.2. パラメタ

「3.2.9 extract\_column\_rr コマンド」を参照のこと。

#### 3.2.17.3. 記述例

'a' を含んだ列から '2' を含んだ列の範囲を抽出する。



```
$オブジェクト名->remove_column_rr(['a', '2']);
```

### 3.3. 抽出指示コマンド

指示したタイミングで抽出を行う。

#### 3.3.1. 書式

\$オブジェクト名->execute();

#### 3.3.2. パラメタ

なし。

#### 3.3.3. 復帰値

終了状態	復帰値
正常終了	0
異常終了	99

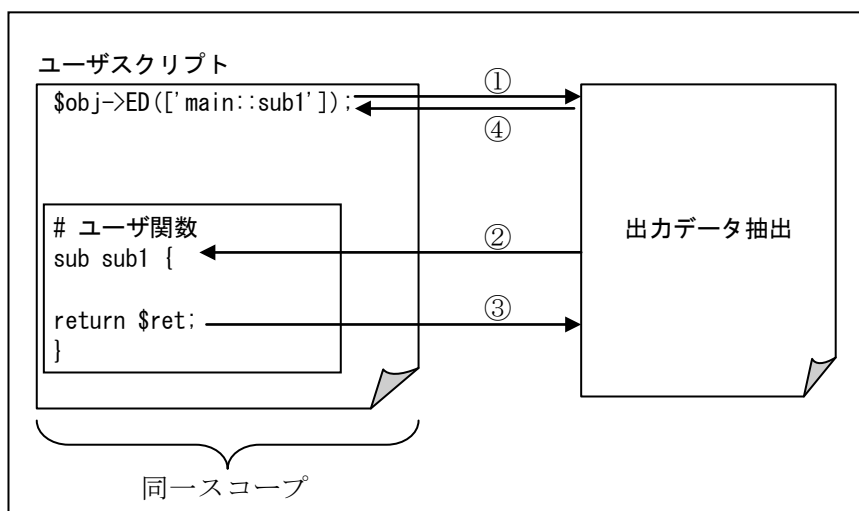
## 第4章 ユーザ関数

出力データ抽出は、当仕様に基いて作成されているユーザ関数のみ動作させることができる。また、ユーザ抽出結果を反映することができる。

尚、ユーザ関数は、当インターフェース仕様通りに作成されていることを前提としている為、出力データ抽出ではユーザ関数の処理や結果に対して一切チェックを行わない。

### 4.3.1. ユーザ関数の記述場所

ユーザ関数は、ユーザスクリプト(メインパッケージ)への記述を基本とする。



スコープが異なる為、推奨はしないが、ユーザスクリプトが読込むパッケージへもユーザ関数を記述することができる。

#### 4.3.2. ユーザ関数の呼出し

ユーザ関数呼出し時、以下の情報を引数として渡す。

- ・ 処理中のレコード情報
- ・ ED コマンドのユーザ引数

&パッケージ::サブルーチン(処理中レコード情報, ユーザ引数…);
------------------------------------

#### 4.3.3. 抽出対象ファイルのデータの読み込み・書き込み等

ユーザ関数内で使用できる関数を用意する。抽出対象ファイルに関しては、ユーザ自身による OPEN・READ・CLOSE を不要とする。

関数の詳細は、「第5章 ユーザ関数内で使える関数」を参照のこと。

#### 4.3.4. ユーザ関数が返却する抽出結果

ユーザ関数は、抽出結果をユーザ関数の復帰値として返却する。抽出結果は、抽出有無を格納した配列形式とし、ユーザ関数内でユーザが自ら定義・設定する。

配列形式

配列	0	1	2	3	...	n
設定内容	行抽出有無	列抽出有無				
		1 列目	2 列目	3 列目	...	n 列目
設定値	{0 1}	[ <u>0</u>  1]	[ <u>0</u>  1]	[ <u>0</u>  1]	...	[ <u>0</u>  1]

例 1) 3 行目を抽出

```

sun user_sub {
    my @ret = ();
    my $num = &get_line_num();
    if ($num eq 3) {
        $ret[0] = 1;          ...行抽出を指定
    }
    return @ret;             ...配列を返却
}

```

例 2) 3 列目を抽出

```

sun user_sub {
    my @ret = ();
    :
    $ret[3] = 1;              ... 3 列目抽出を指定
    return @ret;             ...配列を返却
}

```

## 第5章 ユーザ関数内で使える関数

下記の関数を用意する。

関数名	用途
get_line_num	カレント行の行番号を取得
seek_line	指示行へカレント行を移動
get_line	カレント行データの取得
add_data	カレント行の更新・追加
del_data	カレント行の削除

### 5.1. get\_line\_num

カレント行の行番号を取得する。

#### 5.1.1. 書式

```
$変数名 = &get_line_num();
```

#### 5.1.2. パラメタ

なし。

#### 5.1.3. 記述例

5行目だったら、何らかの処理を行う。

```
sun user_sub {
    my $num = &get_line_num();
    if ($num eq 5) {
        :
    }
}
```

## 5.2. seek\_line

指示行へカレント行を移動する。

### 5.2.1. 書式

```
&seek_line(行番号);
```

### 5.2.2. パラメタ

- 行番号  
移動先の行番号。

### 5.2.3. 記述例

5 行目だったら、1 行前へカレントを移動させる。

```
sun user_sub {  
    my $num = &get_line_num();  
    if ($num eq 5) {  
        &seek_line($num - 1);  
        :  
    }  
}
```



### 5.3. get\_line

カレント行データの取得を行う。

#### 5.3.1. 書式

```
$変数名 = &get_line();
```

#### 5.3.2. パラメタ

なし。

#### 5.3.3. 記述例

5 行目だったら、1 行前のデータを取得する。

```
sun user_sub {  
    my $num = &get_line_num();  
    if ($num eq 5) {  
        &seek_line($num - 1);  
        my $line = &get_line();  
        :  
    }  
}
```

## 5.4. add\_data

カレント行の更新・追加を行う。

### 5.4.1. 書式

```
&add_data("データ");
```

### 5.4.2. パラメタ

- データ  
カレント行へ更新・追加するデータ。

### 5.4.3. 記述例

例1) 5行目に"update"の文字が存在した場合、1行前のデータを取得して更新する。

```
sun user_sub {
  my $line = $_[0];
  my $num = &get_line_num();
  if ($num eq 5 and ($line =~ /update/)) {
    &seek_line($num - 1);
    my $line2 = &get_line();
    &add_data("update $line2");
    :
  }
}
```

例2) 5行目に"add"の文字が存在した場合、"abc"行を追加する。

```
sun user_sub {
  my $line = $_[0];
  my $num = &get_line_num();
  if ($num eq 5 and ($line =~ /add/)) {
    &add_data("$line");
    &add_data("abc");
    :
  }
}
```

## 5.5. del\_data

カレント行データの削除を行う。

### 5.5.1. 書式

```
&del_data();
```

### 5.5.2. パラメタ

なし。

### 5.5.3. 記述例

ある条件に一致する行を削除する。

```
sun user_sub {  
    if (xxxxxxx) {  
        &del_data();  
    }  
}
```