NAME

rend — split data tokens from specified spines into component sub-tokens

SYNOPSIS

rend [**-s**] **-i** *'target_interp'* **-f** *reassign-file* [*inputfile* ...]

DESCRIPTION

The **rend** command breaks apart data tokens from selected input spines into one or more sub-tokens distributed across one or more newly created output spines. The user specifies which input spine or spines are to be split. The manner in which the signifiers are to be distributed is specified in a separate *reassignment* file.

Humdrum data tokens often contain more than one type of information or type of signfier. For example, the `**pitch` representation consists of three parts:† the pitch letter name, the accidental, and the octave number (e.g. A#4). In some tasks it may be useful to split such information into separate spines. For example, a user may wish to reformat the following spine:

```
**pitch
Ab3
Eb4
F#4
C5
*-
```

as three independent spines:

```
**octave    **note    **accidental
3           Ab        b
4           Eb        b
4           F#        #
5           C         .
*-          *-        *-
```

The **rend** command allows each occurrence of a target exclusive interpretation to be replaced by specified output spines. The user selects how the signfiers (characters) in the input spines are to be distributed to the replacement output spines. Signifiers (ASCII characters) are identified using UNIX regular expression syntax (see **regexp (6)**).

† Not including cents deviation.

The above transformation may be achieved by invoking the following command:

```
rend -i '**pitch' -f reassign
```

The **-i** option specifies the target input interpretation, i.e., the input spine(s) to be processed. The **-f** option specifies a reassignment-file (named `reassign`) containing the following records:

```
**octave        [0-9]
**note          [A-Gb#x]+
**accidental    [b#x]+
```

Reassignment files consist of one or more records, each containing two strings separated by a tab. The left-most string identifies the name of the new spine to be generated. The right-most string defines an associated regular expression. Any input signifiers matching the regular expression will be echoed as output in the associated spine. In the above case, all numbers are echoed in the first spine (`**octave`), all letters plus the sharp (#) and flat (b) signs are echoed in the second spine (`**note`), whereas only sharp and flat signs are echoed in the third spine (`**accidental`). The order of the output spines preserves the order of the assignments in the reassignment file. In the above case, for example, the order of the output spines will be `**octave, **note, **accidental` for each input spine labelled `**pitch`.

OPTIONS

The **rend** command provides the following options:

-f *reassign*	maps input tokens to output tokens according to definitions given in the file *reassign*
-h	displays a help screen summarizing the command syntax
-i *target_interp*	process all input spines whose exclusive interpretations are labelled *target_interp*
-s	matches a single instance of the given pattern rather than all instances

Options are specified in the command line.

EXAMPLES

Consider the following example:

```
rend -i '**kern' -f noterest song01
```

and the associate reassignment file named `noterest`:

```
**notes         []A-Ga-g[#-]+|^=+[0-9]*
**rests         [\.0-9r]+|^=+[0-9]*
```

This command specifies that each `**kern` spine in the file `song01` is to be split into two new spines dubbed `**notes` and `**rests`. The first regular expression — `'[]A-Ga-`

`g[#-]+|^=+[0-9]*'` — indicates that the following strings should be echoed in the data records for `**notes`: the upper-case letters `A` to `G` and the lower-case letters `a` to `g`, plus the characters `[`, `]`, `#`, and `-`. Alternatively, **rend** will echo any data token beginning with one or more equals-signs, followed by zero or more numbers.

Similarly, the second `**rests` spine will contain characters that match the regular expression '[\0-9r]+|^+=[0-9]*'. This includes the period (.), all numbers (0-9), plus the letter *r*. Alternatively, **rend** will echo any data token beginning with one or more equals-signs, followed by zero or more numbers.

Given this command, the following input:

```
**kern    **lyrics    **kern
!! Commented input.
8.G       Hi-         4r
16G#      de-         .
=23       =23         =23
8A        ho-         2r
[8c       .           .
8c]       .           .
16r       .           .
16A       .           .
=24       =24         =24
2C 2E     hum.        2r
===       ===         ===
*-        *-          *-
```

will produce the following output:

```
**notes   **rests   **lyrics   **notes   **rests
!! Commented input.
G         8.        Hi-        .         4r
G#        16        de-        .         .
=23       =23       =23        =23       =23
A         8         ho-        .         2r
[c        8         .          .         .
c]        8         .          .         .
.         16r       .          .         .
A         16        .          .         .
=24       =24       =24        =24       =24
C E       2 2       hum.       .         2r
===       ===       ===        ===       ===
*-        *-        *-         *-        *-
```

Notice that **rend** correctly handles Humdrum multiple-stops (such as `2C 2E`). Notice also that if no match is made, a null token (.) is output.

PORTABILITY

DOS 2.0 and up, with the MKS Toolkit. OS/2 with the MKS Toolkit. UNIX systems supporting the *Korn* shell or *Bourne* shell command interpreters, and revised *awk* (1985).

SEE ALSO

cleave (4), **extract** (4), **humsed** (4), **recode** (4), **regexp** (4), **regexp** (6)

WARNINGS

Note that, apart from spine-path terminators, no other spine-path indicators are permitted in spines containing the target interpretation.

BUGS

If the interpretation targetted for processing appears in a spine starting with a different interpretation, the output will fail to generate the proper spine terminator and add-spine path indicators. The result is a non-Humdrum file. Consider the following command:

```
rend -i '**ex1' -f reassign input
```

and the associate reassignment file (`reassign`):

```
**let  [a-z]
**num  [0-9]
```

Given the following input:

```
**ex1    **ex2
a1       b2
*tand1   *tand2
c3       d4
*        **ex1
e5       f6
*-       *-
```

the corresponding output is given below. Note the absence of appropriate spine-path indicators between lines 4 and 5 (hence the output is non-Humdrum).

```
**let    **num    **ex2
a        1        b2
*tand1   *tand1   *tand2
c        3        d4
*        *        **let    **num
e        5        f        6
*-       *-       *-       *-
```