NAME

scramble — randomize order of either Humdrum data records or data tokens

SYNOPSIS

scramble -r [**-s** *regexp*] [*inputfile*] **scramble -t** [**-m**] [**-s** *regexp*] [*inputfile*]

DESCRIPTION

The **scramble** command can be used to randomize either the order of Humdrum data records or the arrangement of data tokens within each data record. The **scramble** command is useful for generating control data when testing contextual relationships between signifiers.

Two modes of operation are supported according to whether the **-r** (record) or **-t** (token) option is invoked. When the **-r** mode is selected, the order of data records is randomized. In this mode, each output record is identical to some input record; only the order of the output records is changed. When the **-t** mode is selected, the order of data tokens within each record is randomized. In this mode, the order of the input records is preserved — however, data tokens between concurrent spines are randomly swapped. The **-t** mode will also cause sub-tokens within multiple stops to be rearranged within the data token. However, if the **-m** option is concurrently invoked, then sub-tokens within multiple-stops will be randomly redistributed across all tokens in the record. The **-r** and **-t** options cannot be invoked concurrently.

In both modes of operation, Humdrum comments and interpretations remain unaffected. Comments and interpretations are output intact, and in precisely the same location (line number) as in the input. Only data records are affected by **scramble.**

Each time **scramble** is invoked, a different random ordering is generated.

Note that when using the **-r** mode, the scrambling process may produce an output that is no longer syntactically correct Humdrum. With the **-r** mode, **scramble** is guaranteed to produce Humdrum output only if (1) the input file is bona fide Humdrum, and (2) the number of spines in the input does not vary.

With the **-t** mode, **scramble** will always produce output conforming to the Humdrum syntax, however it can produce uninterpretable output if concurrent spines do not contain the same type of data (i.e. have the same exclusive interpretations).

Notice also that reordering data records may destroy data-token links such as `**kern` ties.

A skip option (**-s**) permits users to anchor certain data records so that they are not either repositioned, or their data tokens rearranged.

OPTIONS

The **scramble** command provides the following option:

-h	displays a help screen summarizing the command syntax
-m	redistribute subtokens in multiple stops across all tokens in the record; used in conjunction with **-t** only
-r	scramble the order of data records; don't scramble data tokens
-s *regexp*	skip; don't scramble records matching *regexp*; leave matching records intact, and in the same position
-t	scramble data tokens within each record; don't scramble record order

Options are specified in the command line. One of either the *record mode* (**-r**) or *token mode* (**-t**) must be invoked.

EXAMPLES

The use of the **scramble** command can be illustrated using the following input:

```
!! A global comment
!! Another comment
**inter     **inter
*abcd       *efgh
=1          =1
1a 1b       a
!local      !local
2           b1 b2 b3
3           c
=2          =2
!! A later comment.
4a 4b       d
=3          =3
5           e
*-          *-
```

When processed using the *record mode*, the command:

```
scramble -r -s = inputfile
```

might produce the following output:

```
!! A global comment
!! Another comment
**inter     **inter
*abcd       *efgh
=1          =1
3           c
!local      !local
5           e
1a 1b       a
=2          =2
!! A later comment.
4a 4b       d
=3          =3
2           b1 b2 b3
*-          *-
```

In this example, notice that the Humdrum comments and interpretations remain in their original location; only the data records have been reordered. In addition, data records containing an equals-sign have been frozen in their original locations.

When processed using the *token mode*, the command:

```
scramble -t -m -s = inputfile
```

might produce the following output:

```
!! A global comment
!! Another comment
**inter      **inter
*abcd        *efgh
=1           =1
1b           a 1a
!local       !local
b2           b3 2 b1
c            3
=2           =2
!! A later comment.
4a 4b        d
=3           =3
5            e
*-           *-
```

Notice that a complete scrambling of data tokens within a Humdrum file cannot be achieved merely by invoking one **scramble** mode followed by the other mode. In order to completely scramble a Humdrum file the user must extract and scramble the record order for each spine independently, and then reassemble the scrambled spines into a new file using **the assemble** command.

Note also that where the number of spines changes over the course of the input file, valid

Humdrum output is unlikely. Outputs consistent with the Humdrum syntax can be ensured by using the **fields -i** command to chronicle changing numbers of spines in a file, followed by the **yank** command to segregate data blocks containing the same number of spines. Each such bock can be **scramble**d independently and then the blocks reconnected using the UNIX **cat** command. Unnecessary (duplicate) interpretations can be eliminated using **rid -u.**

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

assemble (4), **extract** (4), **fields** (4), **humdrum** (4)

WARNINGS

If the number of spines changes over the course of the input file, valid Humdrum output is unlikely when using the **-r** option. Note also that reordering data records or data tokens will destroy data-token links such as `**kern` "ties." Use of the **-t** mode, can produce uninterpretable output when concurrent spines do not contain the same interpretations.