

---

## NAME

**cleave** — join tokens from two or more spines into a single output spine

## SYNOPSIS

**cleave** [-r] [-d *delim*] -i '\*\*in\_interp1[,\*\*in\_interp2 ...]' -o '\*\*out\_interp' [*inputfile* ...]

## DESCRIPTION

The **cleave** command permits concurrent data tokens in two or more specified spines to be amalgamated into a single data token and output in a single new spine. Consider, for example, two input spines containing pitch information and duration information respectively; **cleave** can be used to form a new spine that combines the pitch and duration signifiers into a single representation.

When the **cleave** command is invoked, the user identifies the input spines to be combined and specifies the name of the resulting output spine. Only a single output spine can be generated by **cleave**.

If necessary, the user may specify delimiter characters that are inserted between the component parts of the combined data token. (See EXAMPLES below.)

The **cleave** command amalgamates all spines containing the interpretation(s) specified by the user. For example, the command:

```
cleave -i '**recip,**kern' -o '**output' katsumi
```

will amalgamate all spines in the file *katsumi* containing **\*\*recip** or **\*\*kern** data — and will output the amalgamated data in a spine labelled **\*\*output**.

Note that the output spine generated by **cleave** preserves the same record-type structure as the input, and so may readily be pasted with the input file using the Humdrum **assemble** command.

The **cleave** command is able to adapt to spine-path changes throughout the input. When a processed input spine is *split*, the new spine participates in the amalgamated output spine. When a processed spine is *exchanged*, **cleave** continues to track its location. When a new spine is *added*, to the input it is included in the amalgamated output only if its interpretation matches one of the interpretations being processed. When a processed spine is *terminated*, **cleave** continues to process other spines in the input; if no target spine remains, then null tokens are generated until the end of the input or until other target interpretations appear. *Join-spine* indicators have no affect on the output.

Notice that the **cleave** command is useful for transforming a spine that periodically splits

and joins into a single spine containing multiple-stops.

## OPTIONS

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

- d** *delimiter*      interpose the string *delimiter* between amalgamated tokens
- i** '*in\_interp*'      list of input interpretations to be processed
- o** '*out\_interp*'    specify output interpretation
- r**                    suppress outputting of duplicate (repeated) signifiers

Options are specified in the command line.

The **-d** option is used to specify a string that is inserted between each component part forming the assembled output token. The default delimiter is the null string.

A given signifier (character) may be present in two or more concurrent input tokens — such as the letter 'A' shared by the tokens 'AB' and 'AX'. Depending on the task, the user may want only a single instance of each signifier to be echoed in the output (e.g. 'ABX' rather than 'ABAX'). The **-r** option causes characters that are common to two or more input tokens to be output only once. The output position of any repeated character corresponds to the first instance of the character in the processed input.

## EXAMPLES

The following examples illustrate the operation of the **cleave** command.

Consider the following input consisting of four spines representing octave-class, diatonic pitch letter name, accidental, and cents-deviation:

**oct	**diaton	**accid	**Cdev
4	G	.	.
4	A	.	.
4	B	b	-10
5	C	.	.
5	D	.	.
5	E	b	.
5	F	#	+12
5	G	.	.
*-	*-	*-	*-

The information available in these four spines might be amalgamated into a single spine by executing the following command:

```
cleave -i '**diaton,**accid,**oct,**Cdev' -o '**pitch' input
```

The following output would be produced:

```

**pitch
G4
A4
Bb4-10
C5
D5
Eb5
F#5+12
G5
*-

```

The output interpretation has been specified as `**pitch`. Notice that the order of the signifiers in each output data token reflects the order of the input interpretations given in the command line — i.e., `**diaton` values first, followed by `**accid`, followed by `**oct`, followed by `**Cdev`. In the case of `**accid` and `**Cdev` data, notice that null tokens (periods) do not affect the output token. In the default invocation, note that no intervening characters are placed between the joined subtokens.

In the following example, **cleave** is used to create double-stops from two spines having identical interpretations. Notice the presence of barlines.

```

**kern    **kern
*M2/4     *M2/4
*foo      *bar
=1        =1
4c        4e
8d        8f
8e        8g
=2        =2
8f        8a
8g        8b
8a        8cc
8b        8dd
=3        =3
4cc       4ee
*-        *-

```

Executing the command

```
cleave -i '**kern' -d ' ' -o '**kern' input > output
```

will produce the following output:



```

**kern
*M2/4
*
=1 =1
4c 4e
8d 8f
8e 8g
=2 =2
8f 8a
8g 8b
8a 8cc
8b 8dd
=3 =3
4cc 4ee
*_

```

Notice that if identical tandem interpretations appear in the target spines, then they are echoed in the output. Otherwise a null interpretation is output.

The redundant measure numbers in the above output might be eliminated using the following **humshed** command:

```
humshed '/^=/s/ .*//' input > output
```

Alternatively, the input might have been preprocessed so that the barlines in one of the two input spines were replaced by null tokens.

The **-r** option can be used to eliminate duplicate or repeated signifiers. Consider, for example, the following input:

**kern	**kern
.	4c
(8	8d
)8	8e
(	8f
)	8g
8'	8a
8'	8b
.	4cc
*_	*_

The first **\*\*kern** spine includes articulation information not present in the second spine. The pitch, duration, and articulation information can be amalgamated without duplication of the duration information using the **-r** option:

```
cleave -r -i '**kern' -o '**kern' input > output
```

The resulting output is:

```

**kern
4c
(8d
)8e
(8f
)8g
8'a
8'b
4cc
*-

```

Users should be careful when using the **-r** option while at the same time assigning a delimiter that appears in the input stream. For example, if the slash (/) is defined as an output delimiter, and the **-r** option is invoked, then following input:

```
ab  a/b
```

will produce the following output:

```
ab//
```

Note that the first slash in the above output delimits the material originally contained in the left and right spines. The second slash is a bona fide signifier in the right spine. If the delimiter in the above example was a space rather than a slash, then the result would produce trailing spaces — and so the output would no longer conform to the Humdrum syntax.

## 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), **humshed** (4), **rend** (4)

## WARNINGS

Syntactically correct Humdrum output is not guaranteed if the **-r** option is invoked while using the space as a delimiter.

The use of regular expression metacharacters as delimiters (such as ^) can cause problems for **cleave**.