

NAME

fribidi_reorder_line - reorder a line of logical string to visual

SYNOPSIS

```
#include <fribidi.h>
```

```
FriBidiLevel fribidi_reorder_line
```

```
(  
    FriBidiFlags flags,  
    const FriBidiCharType *bidi_types,  
    const FriBidiStrIndex len,  
    const FriBidiStrIndex off,  
    const FriBidiParType base_dir,  
    FriBidiLevel *embedding_levels,  
    FriBidiChar *visual_str,  
    FriBidiStrIndex *map  
);
```

PARAMETERS

FriBidiFlags flags

Reorder flags.

const FriBidiCharType *bidi_types

Input list of bidi types as returned by fribidi_get_bidi_types().

Since there are 23 possible values for a **const FriBidiCharType**, they are not all listed here.

const FriBidiStrIndex len

Input length of the line.

const FriBidiStrIndex off

Input offset of the beginning of the line in the paragraph.

const FriBidiParType base_dir

Resolved paragraph base direction.

Possible values for a **const FriBidiParType** are as follows:

FRIBIDI_PAR_LTR	Left-To-Right paragraph.
FRIBIDI_PAR_RTL	Right-To-Left paragraph.
FRIBIDI_PAR_ON	DirectiOn-Neutral paragraph.

FRIBIDI_PAR_WLTR Weak Left To Right paragraph.
FRIBIDI_PAR_WRTL Weak Right To Left paragraph.

FriBidiLevel *embedding_levels

Input list of embedding levels, as returned by `fribidi_get_par_embedding_levels`.

FriBidiChar *visual_str

Visual string to reorder.

FriBidiStrIndex *map

A map of string indices which is reordered to reflect where each glyph ends up.

DESCRIPTION

This function reorders the characters in a line of text from logical to final visual order. This function implements part 4 of rule L1, and rules L2 and L3 of the Unicode Bidirectional Algorithm available at http://www.unicode.org/reports/tr9/#Reordering_Resolved_Levels.

As a side effect it also sets position maps if not NULL.

You should provide the resolved paragraph direction and embedding levels as set by `fribidi_get_par_embedding_levels()`. Also note that the embedding levels may change a bit. To be exact, the embedding level of any sequence of white space at the end of line is reset to the paragraph embedding level (That is part 4 of rule L1).

Note that the bidi types and embedding levels are not reordered. You can reorder these (or any other) arrays using the map later. The user is responsible to initialize map to something sensible, like an identity mapping, or pass NULL if no map is needed.

There is an optional part to this function, which is whether non-spacing marks for right-to-left parts of the text should be reordered to come after their base characters in the visual string or not. Most rendering engines expect this behavior, but console-based systems for example do not like it. This is controlled by the `FRIBIDI_FLAG_REORDER_NSM` flag. The flag is on in `FRIBIDI_FLAGS_DEFAULT`.

RETURNS

Maximum level found in this line plus one, or zero if any error occurred (memory allocation failure most probably).

SEE ALSO

`fribidi_shape_arabic(3)`, `fribidi_get_par_direction(3)`, `fribidi_get_par_embedding_levels_ex(3)`,

fribidi_get_bidi_type(3), fribidi_get_bidi_types(3), fribidi_get_bidi_type_name(3),
fribidi_debug_status(3), fribidi_set_debug(3), fribidi_charset_to_unicode(3),
fribidi_unicode_to_charset(3), fribidi_parse_charset(3), fribidi_mirroring_status(3),
fribidi_set_mirroring(3), fribidi_reorder_nsm_status(3), fribidi_set_reorder_nsm(3),
fribidi_log2vis_get_embedding_levels(3), fribidi_get_type(3), fribidi_get_type_internal(3),
fribidi_get_par_embedding_levels(3), fribidi_join_arabic(3), fribidi_get_joining_type(3),
fribidi_get_joining_types(3), fribidi_get_joining_type_name(3), fribidi_get_mirror_char(3),
fribidi_shape_mirroring(3), fribidi_get_bracket(3), fribidi_get_bracket_types(3), fribidi_shape(3),
fribidi_remove_bidi_marks(3), fribidi_log2vis(3)