

SetWordBreak

Install a custom "word-break" routine

#include <TextEdit.h>

TextEdit

```
void      SetWordBreak(wbProc, hTE );
ProcPtr   wbProc ;           address of your custom routine
TEHandle   hTE ;             handle of an edit record
```

SetWordBreak lets you customize how **TextEdit** will decide where to wrap from one screen line to the next. It also affects how much text constitutes the "word" the user selects text with a double-click.

wbProc is the address of your custom word-breaking routine. Use NIL (0) to revert to the standard word breaker.

hTE is a handle obtained via **TENew** or **TEStylNew**. It leads to a variable-length TERec structure and identifies the edit record to be affected by this change.

Returns: none

Notes: By default, **TextEdit** considers a "word" to be an island of text surrounded by characters whose ASCII values are 0x20 or less (i.e., the space character or non-printing control characters such as 0x0D). By installing a custom routine, you can force **TextEdit** to break words at punctuation such as periods, commas, parentheses, and so forth or allow words to contain special characters such as , , and .

Your custom routine should be declared as:

```
pascal Boolean myWordBreaker(Ptr textPtr, short offset )
{
    if ( textPtr[offset] ... is a break character ... )
        return( TRUE );
    else
        return( FALSE );
}
```

Install the routine via:

```
SetWordBreak( myWordBreaker, hTE );
```

Or, just store the address into the TERec structure:

```
(*hTE)->wordBreak=myWordBreaker;
```

You may also wish to install a custom line-calculating routine by storing an address in the global variable TERecal. It will let you manipulate the line-start offsets in a TERec. See **TECalText** for related information.