TEContinuousStyle Check if a style element is continuous across selection

#include <TextEdit.h> **TextEdit**

```
Boolean     TEContinuousStyle(&mode, &theStyle, hTE );
short       *mode ;        address of 16-bit Style Mode; recieves coded data
TextStyle   *theStyle ;    address of a 12-byte record containing style info
TEHandle    hTE ;          edit record of interest
            returns        Is specified attribute continuous?
```

The **TEContinuousStyle** function, new with System 6.0, gives you information about the attributes of the current selection.

TEContinuousStyle examines the current selection range and checks if a specified style attribute is continuous across the current selection range. You can use this as an aid in toggling styles (see **TESetStyle**) or to determine which, if any, items in your Style... menu should have a check mark.

mode is the address of a short. On entry, it specifies a style operation mode (see **Style Mode**). Bits of this value specify which characteristics of the selected text should be examined.

Upon return, each bit that was set on entry is been cleared if that style element was not continuous.

theStyle is the address of a 12-byte TextStyle structure. On entry, it identifies which characteristics to examine. Upon exit, fields corresponding to set-bits in mode are filled-in to reflect the values of any attributes which are continuous.

hTE is a handle leading to an edit record created via **TEStylNew**.

The mode parameter, which takes the same values as in **TESetStyle**, specifies which attributes should be checked. When **TEContinuousStyle** returns, the mode parameter indicates which of the checked attributes is continuous over the selection range, and the *aStyle* parameter reflects the continuous attributes.

TEContinuousStyle returns TRUE if all of the attributes to be checked are continuous and returns FALSE if they are not. In other words, if the mode parameter is the same before and after the call, then **TEContinuousStyle** returns TRUE.

Listing below illustrates how **TEContinuousStyle** is useful for marking the Style menu items so they correspond to the current selection.

Marking the Style menu items so they correspond to the current selection

```
short     mode;
TextStyle aStyle;
TEHandle  myTE;
MenuHandle    styleMenu;
```

```
mode = doFace;

if (TEContinuousStyle(&mode, &aStyle, myTE)) {
    // There is at least one face that is continuous over
    // the selection. Note that it might be plain, which is
    // actually the absence of all styles

    CheckItem(styleMenu, plainItem, aStyle.tsFace == normal);
    CheckItem(styleMenu, boldItem, aStyle.tsFace == bold);
    CheckItem(styleMenu, italicItem, aStyle.tsFace == italic);

    // Set other menu items appropriately
}
else {
    // No text face is common to the entire selection.
    CheckItem(styleMenu, plainItem, FALSE);
    CheckItem(styleMenu, boldItem, FALSE);
    CheckItem(styleMenu, italicItem, FALSE);
    // Set other menu items appropriately.
}
```

You can also use **TEContinuousStyle** to determine the actual values for those attributes that are continuous for the selection. Note that a field in the text style record is only valid if the corresponding bit is set in the mode variable; otherwise, the field contains invalid information. Listing below illustrates how you might use **TEContinuousStyle** to determine the font, face, size, and color of the current selection.

Determining the font, face, size, and color of the current selection

```
short    mode;
Boolean  continuous;
TextStyle aStyle;
TEHandle  myTE;


mode = doFont + doFace + doSize + doColor;
continuous = TEContinuousStyle(&mode, &aStyle, myTE);
if (BitAnd (mode, doFont) != 0) {

    // font for selection = aStyle.tsFont
}
else {

    // more than one font in selection
}

if (BitAnd(mode, doFace) != 0) {

    // aStyle.tsFace contains the text faces (or plain) that
    // are common to the selection
```

```
}
else {

    // No text face is common to the entire selection
}

if (BitAnd(mode, doSize) != 0) {

    // size for selection = aStyle.tsSize
}
else {

    // more than one size in selection
}

if (BitAnd(mode, doColor) != 0) {

    // color for selection = aStyle.tsColor
}
else {

    // more than one color in selection
}
}
```

When **TEContinuousStyle** returns a mode that contains doFace and returns an *aStyle* parameter with a tsFace field of [bold, italic], it means that the selected text is all bold and all italic, but may contain other text faces as well. None of the other faces applies to all of the selected text, or it would have been included in the tsFace field. But if the tsFace field is the empty set, then all of the selected text is plain.

If the current selection range is an insertion point, **TEContinuousStyle** returns the style information for the next character to be typed. **TEContinuousStyle** always returns TRUE in this case, and each field of the text style record is set if the corresponding bit in the *mode* parameter was set. If hTE is a handle to an unstyled edit record, **TEContinuousStyle** returns the simple style information of the entire record.

Returns: a Boolean. It will be on of:

FALSE (0) the style is not continuous

TRUE (0) **all** specified attributes of the style are continuous; it (they) applies to all of the text in the selection range OR selection range is an insertion point.

Notes: Notice that *mode* is the address of a short in this function and that it receives return information. In effect, each bit of *mode* is a Boolean corresponding to the truth of continuousness of an attribute (face, font, size, etc.)

The returned values in *theStyle* tell you the actual values of the attributes which were continuous. For instance, in a call such as:

```
TextStyle       theStyle;
```

```
short           theMode;

theMode =  doFont | doSize | doFace;
theStyle.tsFace = bold | italic | outline;

TEContinuousStyle( &theMode, &theStyle, hTE )
if (theMode & doFont) { ....font now in theStyle.tsFont is continuous ... }
if (theMode & doSize) { ....font size now in theStyle.tsSize is continuous ... }
if (theMode & doFace) { ....face(s) in theStyle.tsFace are continuous ... }
```

Note that bits of *mode* (and corresponding fields of *theStyle*) you did not request will contain garbage. For instance, in the above example, the doColor bit is not set on entry, so the return value of doColor and any returned value in *theStyle*->tsColor will be meaningless.

The return value in *theStyle*->tsFace will reflect only those values that are continuous (other face attributes may exist in the selection, but if so, they are not continuous). However, if the *mode* doFace bit is set on return, and *theStyle*->tsFace is 0, the "plain" text face is continuous across the selection.