

TEFeatureFlag Return last setting of a specified feature's bit

#include <TextEdit.h>

TextEdit

short **TEFeatureFlag**(*feature, action, hTE*);
short *feature* ; enable/disable inline input
short *action* ; like *feature*, using bit constants
TEHandle *hTE* ; handle to the edit record

returns either TEBitSet or TEBitClear

The **TEFeatureFlag** function allows you to enable outline highlighting and text buffering in your application. You can also use this function to disable inline input in a particular edit record and to enable several new features that have been provided so that inline input works correctly with **TextEdit**.

Note: To test for the availability of these features, you can call the **Gestalt** function with the gestaltTextEditVersion selector. A result of gestaltTE4 or greater returned in the response parameter indicates that outline highlighting and text buffering are available. A result of gestaltTE5 or greater returned in the response parameter indicates that the two inline input features are available. (For details, see the description about Determining the Version of TextEdit.)

The inline input features are also available on version 6.0.7 systems with non-Roman script systems installed. However, there is no **Gestalt** constant that indicates this availability.

The feature parameter allows you to disable inline input in a particular edit record or to specify the features you want to enable-outline highlighting, text buffering, and features provided for inline input in **TextEdit**. The action parameter lets you enable and disable these features by using the TEBitSet and TEBitClear constants and lets you test the settings of these feature bits by using the TEBitTest constant. The hTE parameter is a handle to the edit record.

The **TEFeatureFlag** function returns the previous setting of the feature's bit, either TEBitSet or TEBitClear.

Note that there is also a constant named TEFeatureFlag which has the following values:

The feature or bit definitions for TEFeatureFlag are:

teFAutoScr	= 0,	/*00000001b*/
teFTextBuffering	= 1,	/*00000010b*/
teFOutlineHilite	= 2,	/*00000100b*/
teFInlineInput	= 3,	/*00001000b*/
teFUseTextServices	= 4,	/*00010000b*/

The action for the new "bit (un)set" interface, TEFeatureFlag is:

TEBitClear	= 0,
TEBitSet	= 1 /*set the selector bit*/