EqualString See if two Pascal-style strings are equal

#include <OSUtils.h> **Operating System Utilities**

Boolean	**EqualString(***strA*, *strB*, *caseSens*, *diacSens* **);**	
Str255	*strA* ;	Pascal-style strings to compare
Str255	*strB* ;	
Boolean	*caseSens* ;	should upper/lowercase count?
Boolean	*diacSense* ;	should diacritical marks count?
	returns	Are the strings equal?

EqualString compares two pascal-style length-prefixed strings (optionally ignoring case and/or diacritical marks), and returns an indication of whether or not they are equal.

strA and . . .
strB are addresses of Pascal-style length-prefixed strings.

caseSens specifies whether or not the comparison should be case-sensitive. It must be one of:

FALSE	ignore character case when comparing ('A' == 'a')
TRUE	character case is significant ('A' != 'a')

diacSens specifies whether or not the comparison should be sensitive to diacritical marks. It must be one of:

FALSE	ignore diacritical marks when comparing ('å' == 'a')
TRUE	diacritical marks are significant ('å' != 'a')

Returns: a Boolean; it indicates whether the strings are equal, considering the case- and diacritical sensitivity. It is one of:

FALSE	not equal
TRUE	equal

Notes: Since **EqualString** compares pascal-style strings directly, it is handier than converting to C-style strings and using strcmp. Examples:

```
Str255     strA="\pAbcDef";
Str255     strB="\påbcdef";

EqualString( strA,strB, TRUE,TRUE);        /* Returns FALSE */
EqualString( strA, strB, FALSE,FALSE);     /* Returns TRUE */
```

If *caseSens* =FALSE, then both strings are treated as if they had been upshifted with **UprString** (though the original contents are not modified).

The **RelString** function [128K ROMs] is more flexible in that its return code identifies which string is higher or lower in the collating sequence. The **IUEqualString** and **IUCompString** functions take into consideration special spelling conventions used in foreign languages.