```
// Revert the data
CreateSampleData()

// Sort based on text key
SortColumns keyWaves=text1,sortWaves=w1

// Revert the data
CreateSampleData()

// Sort using key index
SortColumns/kndx=0 sortWaves={text1,w1}
```

See Also

Sorting on page III-132, **Sort**, **Reverse**, **SortList**

SortList

SortList(*listStr* [, *listSepStr* [, *options*])

The SortList function returns *listStr* after sorting it according to the default or *listSepStr* and *options* parameters. *listStr* should contain items separated by *listSepStr*, such as "the first item;second item;".

Use SortList to sort the items in a string containing a list of items separated by a string, such as those returned by functions like **TraceNameList** or **WaveList**, or a line of text from a delimited text file, where listSepStr can be "\r" or "\r\n".

listSepStr and *options* are optional; their defaults are ";" and 0 (ascending alphabetic sort), respectively.

Details

listStr is treated as if it ends with a *listSepStr* even if it doesn't. The returned list will always have an ending *listSepStr* string.

In Igor6, SortList used only the first byte of listSepStr. As of Igor7, it uses the whole string.

options controls the sorting method, as follows:

0:	Default sort (ascending case-sensitive alphabetic ASCII sort).
1:	Descending sort.
2:	Numeric sort.
4:	Case-insensitive sort.
8:	Case-sensitive alphanumeric sort.
16:	Case-insensitive alphanumeric sort that sorts wave0 and wave9 before wave10.
32:	Unique sort in which duplicates are removed. Added in Igor Pro 7.00.
64:	Ignore + and - in the alphanumeric sort so that “Text-09” sorts before “Text-10”. Set options to 80 or 81. Added in Igor Pro 7.00.

options may also be a bitwise combination of these values with the following restriction: only one of 2, 4, 8, or 16 may be specified. Thus the legal values are thus 0, 1, 2, 3, 4, 5, 8, 9, 16, 17, 32, 33, 34, 35, 36, 40, 41, 48, 49, 80 or 81. Other values will produce undefined sorting.

In a case-insensitive, unique sort (options=4+32), if two items differ only in case, which one is retained is not specified.

Examples

```
// Alphabetic sorts
Print SortList("c;a;a;b")                    // prints "a;a;b;c;"
Print SortList("you,me,More", ",", 0)        // prints "More,me,you,"
Print SortList("you,me,More", ",", 4)        // prints "me,More,you,"
Print SortList("9,93,91,33,15,3", ",")       // prints "15,3,33,9,91,93,"
Print SortList("Zx;abc;All;", ";", 0)        // prints "All;Zx;abc;"
Print SortList("Zx;abc;All;", ";", 8)        // prints "abc;All;Zx;"
Print SortList("w9;w10;w02;", ";", 16)       // prints "w02;w9;w10;"

// Unique sort
Print SortList("b;c;a;a;", ";", 32)          // prints "a;b;c;"
```