ReplaceNumberByKey

ReplaceNumberByKey(*keyStr*, *kwListStr*, *newNum* [, *keySepStr* [, *listSepStr* [, *case*]]])

The ReplaceNumberByKey function returns *kwListStr* after replacing the numeric value of the keyword-value pair specified by *keyStr*. *kwListStr* should contain keyword-value pairs such as `"KEY=value1,KEY2=value2"` or `"Key:value1;KEY2:value2"`, depending on the values for *keySepStr* and *listSepStr*.

Use ReplaceNumberByKey to add or modify numeric information in a string containing a `"key1:value1;key2:value2;"` style list such as those returned by functions like **AxisInfo** or **TraceInfo**.

If *keyStr* is not found in *kwListStr*, then the key and the value are appended to the end of the returned string.

keySepStr, *listSepStr*, and *case* are optional; their defaults are ":", ";", and 0 respectively.

Details

The actual string appended is:

[*listSepStr*] *keyStr keySepStr newNum listSepStr*

The optional leading list separator *listSepStr* is added only if *kwListStr* does not already end with a list separator.

kwListStr is searched for an instance of the key string bound by *listSepStr* on the left and a *keySepStr* on the right. The text up to the next ";" is replaced by *newNum* after conversion to text using the %.15g format (see **printf** for format conversion specifications).

kwListStr is treated as if it ends with a *listSepStr* even if it doesn't.

Searches for *keySepStr* and *listSepStr* are always case-sensitive. Searches for *keyStr* in *kwListStr* are usually case-insensitive. Setting the optional *case* parameter to 0 makes the comparisons case sensitive.

In Igor6, only the first byte of *keySepStr* and *listSepStr* was used. In Igor7 and later, all bytes are used.

If *listSepStr* is specified, then *keySepStr* must also be specified. If *case* is specified, *keySepStr* and *listSepStr* must be specified.

Examples

```
Print ReplaceNumberByKey("K1", "K1:7;", 4)                // prints "K1:4;"
Print ReplaceNumberByKey("k2", "K2=8;", 5, "=")           // prints "K2=5;"
Print ReplaceNumberByKey("K3", "K3:9,", 6, ":", ",")      // prints "K3:6,"
Print ReplaceNumberByKey("k3", "K0:9", 6, ":", ",")       // prints "K0:9,k3:6,"
Print ReplaceNumberByKey("k3", "K3:9,", 6, ":", ",")      // prints "K3:6,"
Print ReplaceNumberByKey("k3", "K3:9,", 6, ":", ",", 1)  // prints "K3:9,k3:6,"
```

See Also

The **ReplaceStringByKey**, **NumberByKey**, **StringByKey**, **RemoveByKey**, **ItemsInList**, **AxisInfo**, **IgorInfo**, **SetWindow**, and **TraceInfo** functions.

ReplaceString

ReplaceString(*replaceThisStr*, *inStr*, *withThisStr* [, *caseSense* [, *maxReplace*]])

The ReplaceString function returns *inStr* after replacing any instance of *replaceThisStr* with *withThisStr*.

The comparison of *replaceThisStr* to the contents of *inStr* is case-insensitive. Setting the optional *caseSense* parameter to nonzero makes the comparison case-sensitive.

Usually all instances of *replaceThisStr* are replaced. Setting the optional *maxReplace* parameter limits the replacements to that number.

Details

If *replaceThisStr* is not found, *inStr* is returned unchanged.

If *maxReplace* is less than 1, then no replacements are made. Setting `maxReplace = Inf` is the same as omitting it.

Examples

```
Print ReplaceString("hello", "say hello", "goodbye")// prints "say goodbye"
Print ReplaceString("\r\n", "line1\r\nline2", "") // prints "line1line2"
Print ReplaceString("A", "an Ack-Ack", "a", 1)     // prints "an ack-ack"
Print ReplaceString("A", "an Ack-Ack", "a", 1, 1) // prints "an ack-Ack"
Print ReplaceString("", "input", "whatever")   // prints "input" (no change)
```