

StrGet() / StrPut() [v1.0.90+]

Reads/Writes a string from or to a memory address, optionally converting from or to a given code page.

```
StrGet(Address [, Length] [, Encoding])
```

```
StrPut(String [, Encoding])
```

```
StrPut(String, Address [, Length] [, Encoding])
```

Parameters

String

| A string or a number (If it is a number, it will be treated as a string, as if each digit was a character.)

Address

| The address at which a string will be read from or *String* will be written to

Length

| The maximum number of **characters** to read/write, including the null-terminator if required

Encoding

A string, representing the source encoding for StrGet or the target encoding for StrPut; numeric identifiers must be prefixed with "CP".

Specify an empty string ("") or "CP0" to use the system default ANSI code page. However, specifying an empty string ("") is **not** the same as leaving this parameter empty. If *Encoding* is not specified, it defaults to "UTF-16" for Unicode versions of the AutoHotkey executable and to "CP0" for ANSI versions. (See *Return Value* below for more details.)

Examples: "UTF-8", "UTF-16", "CP936", "" or "CP0". A list with with many common code page identifiers, besides other sources, can be found [here](#).

Return Value

To understand the return values, note that the return value of StrGet and the *String* parameter of StrPut are always in the native encoding of the current AutoHotkey executable (Unicode or ANSI), whereas *Encoding* specifies the encoding of the string written to or read from the given *Address*. (Background info can be found under [String Encoding](#) and [Unicode vs ANSI](#) and the links therein.)

If *Encoding* is not specified, it defaults, as mentioned above, to "UTF-16" for Unicode versions of the AutoHotkey executable and to "CP0" for ANSI versions. In effect, this means, that in this case the string is simply processed without any conversion of the encoding taking place. (To determine at runtime which version of AutoHotkey is executing your script, use the build-in variable [A_IsUnicode](#).)

Also important to know is, that, unless manually tempered with, a string in AutoHotkey always includes a null-termination character (also called null-terminator) at its end.

StrGet:

StrGet returns the string read from *Address* either with a length of *Length* characters or up to the first encountered null-terminator, whichever occurs first, after performing any necessary conversion.

The returned string always includes a null-terminator at the end, even if no null-terminator was encountered because of having read *Length* characters first.

Note:

If *Length* is not specified and StrGet does not encounter a null-terminator, it will continue reading characters until running into an unpredictable error.

StrPut:

(A) If *Address* is **not** specified, StrPut returns the required buffer size in **characters**, including the null-termination character.

(B) If at least *String* and *Address* are specified, StrPut returns the number of **characters** written.

If *Length* is exactly the length of the string without the null-terminator, then the string is written to memory without the null-terminator, nor is it included in the returned count.

If *Length* is longer than that or not specified, the string is written to memory with the null-terminator but no other characters thereafter. In this case, the returned count includes the null-terminator.

If *Length* is less than the length of the string without the null-terminator, the function fails and returns 0.

Upon invalid parameters both functions return an empty string. Upon other errors, the both functions return 0.

Remarks

If conversion between code pages takes place, the required buffer size in bytes may differ from the size of the source *String*.

If *Length* is specified, then the prefixed of numeric identifiers with "CP" is optional. In that case the numeric identifiers can even be numbers and don't have to be strings.

Scripts which are required to be compatible with AutoHotkey Basic can still use StrPut and StrGet provided that the appropriate script files are installed in a [function library](#). These scripts can be found at the [archived AutoHotkey Forum](#).

Related

[Concepts and Conventions](#), [Script Compatibility](#), [VarSetCapacity\(\)](#)

Examples

Encoding may be specified directly after *String* or *Address*, but in those cases *Encoding* must be non-numeric:

```
strA := StrGet(addressA, "cp0")      ; OK
strA := StrGet(addressA, lengthL, 0) ; OK
strA := StrGet(addressA, 0)         ; returns an empty string
countC := StrPut(stringS, addressA, lengthL, 0) ; OK
countC := StrPut(stringS, addressA, 0) ; Error
```

StrPut may be called once to calculate the required buffer size for a string in a particular encoding, then again to encode and write the string into the buffer. If you frequently use variables with StrPut, consider adding this function to your [library](#):

```
StrPutVar(string, ByRef var, encoding)
{
    ; Ensure capacity.
    VarSetCapacity( var, StrPut(string, encoding)
        ; StrPut returns char count, but VarSetCapacity needs bytes.
        * ((encoding="utf-16"||encoding="cp1200") ? 2 : 1) ) ; The numeric identifier 1200 is equivalent to "UTF-16"
    ; Copy or convert the string.
    return StrPut(string, &var, encoding)
}
```