VBspeed © 2000-10, updated: 05-Oct-2002

InStr

See also InStrRev

The Definition

Function InStr

Returns the position of the first occurrence of one string within another.
Native to VB5 and VB6 (actually to all versions of VB, even to all versions of BASIC, AFAIK).

VBspeed's Declaration:

```
InStr(sCheck, sMatch[, Start[, Compare]])
```

VB's native Declaration:

VB's InStr, of course, looks like this (with the optional Start parameter as the first argument, a weird setup if you ask me):

```
InStr([Start, ]sCheck, sMatch[, Compare])
```

Arguments:

sCheck Required. String expression being searched.

sMatch Required. String expression being searched for.

Start Optional. Numeric expression that sets the starting position for each search. If omitted, search begins at the first character position (Start = 1).

Compare Optional. Numeric value indicating the kind of comparison to use when evaluating substrings. If omitted, a binary comparison is performed.

Return Values:

```
Typical cases:
InStr("abc", "a")             =>  1
InStr("abc", "b")             =>  2

Special cases:
sCheck is zero-length         =>  0
sMatch is zero-length         =>  start
sMatch is not found           =>  0
Start > Len(sMatch)           =>  0
```

Roll your own

If you want to have a go at **InStr** yourself, use this function (VB5/6-compatible) to verify the correctness of your code.

The Charts

Calls

	lRet = InStr(sCheck, sMatch, , Compare)

Call 1	sCheck = "http://www.xbeat.net/vbspeed/index.htm"
	sMatch = ":"

Call 2	sCheck = "http://www.xbeat.net/vbspeed/index.htm"
	sMatch = "."

Call 3	sCheck = "http://www.xbeat.net/vbspeed/index.htm"
	sMatch = "m"

Call 4	sCheck = "http://www.xbeat.net/vbspeed/index.htm"
	sMatch = "M"
	Compare = **vbTextCompare**

Call 5	sCheck = "http://www.xbeat.net/vbspeed/index.htm"
	sMatch = "www"

Call 6	sCheck = "http://www.xbeat.net/vbspeed/index.htm"
	sMatch = "WWW"
	Compare = **vbTextCompare**

Call 7	sCheck = Space$(999) & String$(99, "x") & Space(99) sMatch = "x" [many occurrences, but very far from left end]

Call 8	sCheck = Space$(100)
	sMatch = "x" [no occurrences]

VB5

Code	Author	Doping	Notes	Call 1			Call 2			Call 3			Call 4			Call 5			Call 6			Call 7			Call 8		
InStr	VB5/6			2	1.08	0.070µs	1	1.00	0.076µs	1	1.00	0.094µs	2	8.07	1.311µs	1	1.00	0.080µs	2	11.06	1.328µs	1	1.00	1.415µs	1	1.00	0.182µs
InStr01	Marzo	API		1	1.00	0.065µs	2	1.25	0.094µs	2	1.36	0.128µs	1	1.00	0.162µs	2	1.21	0.097µs	1	1.00	0.120µs	2	2.30	3.256µs	2	1.99	0.361µs

VB6

Code	Author	Doping	Notes	Call 1			Call 2			Call 3			Call 4			Call 5			Call 6			Call 7			Call 8		
InStr	VB5/6			1	1.00	0.063µs	1	1.00	0.072µs	1	1.00	0.087µs	2	8.22	1.306µs	1	1.00	0.076µs	2	10.67	1.309µs	1	1.00	1.414µs	1	1.00	0.183µs
InStr01	Marzo	API		2	1.02	0.064µs	2	1.28	0.092µs	2	1.40	0.122µs	1	1.00	0.159µs	2	1.19	0.090µs	1	1.00	0.123µs	2	2.17	3.065µs	2	1.76	0.322µs

Conclusions

It's not easy to defeat native InStr, but there's good news for people that have to use InStr in TextCompare mode.

Mail your code!

The Code

InStr01

submitted 05-Oct-2002 by Marzo Junior

Doping: TLB (cf. Dope'n'Declarations)

```
Class-wrapped. The class, which also includes a bunch of related functions, is waiting
for you here.
```

Author's comments:

Donald's comments:

top | charts