BLASM
Create Processor Instructions

Setting up Processors and Segments
Before you can use the instructions, you must set up some segments and some Processors. Segments is needed because the instructions writes the data to them. Processors is also needed for setting instruction compability.

Setting up

```
; The segments
segment "Code Segment" .code        ; The .code-segment
segment "Data Segment" .data        ; The .data-segment
segment "Bss Segment" .bss          ; The .bss (es)-segment
segment "Stack Segment" .sseg       ; The stack segment
segment "F-Segment" .fseg           ; The F-segment (FS)
segment "G-Segment" .gseg           ; The G-segment (GS)
.code                               ; Set .code segment

; Processors
.processor .086 0                   ; 80086
.processor .186 1                   ; 80186
.processor .286 2                   ; 80286
.processor .386 3                   ; 80386
.processor .486 4                   ; 80486
.processor .586 5                   ; 80586
.processor .586p 6                  ; 80586+
.086                                ; Set 80086-processor
```

Instructions
BLASM's initial data source ("\Blasm\IDS\init.bds") is already full of instructions and settings for the **Intel**- and **AMD**-processors. But if you want to make your own instructions, if something is missing or maybe for other processors etc., use the **INSTR**-instruction:

Syntax:

```
instr   name  param1  param2  data
.instr  name  param1  param2  data
```

Parameter	Type	Description
name	name	Name of the instruction
param1	value	Parameter code (1)
param2	value	Parameter code (2)
data	data string	Instruction Data

Parameter Codes

TYPE	VALUE
none	0
8-bit register	1
16-bit register	2
32-bit register	3
Segment register	4
Control register	5
Debug register	6
Test register	7
AL register	8
AX register	9
EAX register	10
CL register	11
DX register	12
SI register	13
DI register	14
BX register	15
BYTE PTR [SI]	16
BYTE PTR [DI]	17
BYTE PTR [BX]	18
WORD PTR [SI]	19
WORD PTR [DI]	20
WORD PTR [BX]	21
DWORD PTR [SI]	22
DWORD PTR [DI]	23
DWORD PTR [BX]	24
ES register	25
CS register	26
SS register	27
DS register	28
FS register	29
GS register	30
ES:SI	31
ES:DI	32
ES:BX	33
CS:SI	34
CS:DI	35
CS:BX	36
SS:SI	37
SS:DI	38
SS:BX	39
DS:SI	40
DS:DI	41
DS:BX	42
FS:SI	43
FS:DI	44
FS:BX	45
GS:SI	46
GS:DI	47
GS:BX	48
ES:memory	49
CS:memory	50
SS:memory	51
DS:memory	52
FS:memory	53
GS:memory	54
ES:value	55
CS:value	56
SS:value	57
DS:value	58
FS:value	59
GS:value	60

Name	61
BYTE memory	62
WORD memory	63
DWORD memory	64
Value	65
BYTE value	66
WORD value	67
DWORD value	68
Value = "1"	69
"@CODE"	70
"@DATA"	71
"@SSEG"	72
"@BSS"	73
"@FSEG"	74
"@GSEG"	75
"INS"	76
"INSB"	77
"INSW"	78
"INSD"	79
"MOVS"	80
"MOVSB"	81
"MOVSW"	82
"MOVSD"	83
"OUTS"	84
"OUTSB"	85
"OUTSW"	86
"OUTSD"	87
"LODS"	88
"LODSB"	89
"LODSW"	90
"LODSD"	91
"STOS"	92
"STOSB"	93
"STOSW"	94
"STOSD"	95
"CMPS"	96
"CMPSB"	97
"CMPSW"	98
"CMPSD"	99
"SCAS"	100
"SCASB"	101
"SCASW"	102
"SCASD"	103
BX + SI	104
BX + DI	105
BP + SI	106
BP + DI	107
BX + SI + memory	108
BX + DI + memory	109
BP + SI + memory	110
BP + DI + memory	111
SI + memory	112
DI + memory	113
BP + memory	114
BX + memory	115
BX + SI + value	116
BX + DI + value	117
BP + SI + value	118
BP + DI + value	119
SI + value	120
DI + value	121
BP + value	122
BX + value	123

DL register	124
BL register	125
AH register	126
CH register	127
DH register	128
BH register	129
CX register	130
SP register	131
BP register	132
ECX register	133
EDX register	134
EBX register	135
ESP register	136
EBP register	137
ESI register	138
EDI register	139
CR0 register	140
CR2 register	141
CR3 register	142
CR4 register	143
DR0 register	144
DR1 register	145
DR2 register	146
DR3 register	147
DR6 register	148
DR7 register	149
TR3 register	150
TR4 register	151
TR5 register	152
TR6 register	153
TR7 register	154
[SI]	155
[DI]	156
[BX]	157
[name]	158
[value]	159
[ES:BX+SI]	160
[ES:BX+DI]	161
[ES:BP+SI]	162
[ES:BP+DI]	163
[ES:BX+SI+Memory]	164
[ES:BX+DI+Memory]	165
[ES:BP+SI+Memory]	166
[ES:BP+DI+Memory]	167
[ES:SI+Memory]	168
[ES:DI+Memory]	169
[ES:BP+Memory]	170
[ES:BX+Memory]	171
[ES:BX+SI+Value]	172
[ES:BX+DI+Value]	173
[ES:BP+SI+Value]	174
[ES:BP+DI+Value]	175
[ES:SI+Value]	176
[ES:DI+Value]	177
[ES:BP+Value]	178
[ES:BX+Value]	179

[CS:BX+SI]	180
[CS:BX+DI]	181
[CS:BP+SI]	182
[CS:BP+DI]	183
[CS:BX+SI+Memory]	184
[CS:BX+DI+Memory]	185
[CS:BP+SI+Memory]	186
[CS:BP+DI+Memory]	187
[CS:SI+Memory]	188
[CS:DI+Memory]	189
[CS:BP+Memory]	190
[CS:BX+Memory]	191
[CS:BX+SI+Value]	192
[CS:BX+DI+Value]	193
[CS:BP+SI+Value]	194
[CS:BP+DI+Value]	195
[CS:SI+Value]	196
[CS:DI+Value]	197
[CS:BP+Value]	198
[CS:BX+Value]	199
[SS:BX+SI]	200
[SS:BX+DI]	201
[SS:BP+SI]	202
[SS:BP+DI]	203
[SS:BX+SI+Memory]	204
[SS:BX+DI+Memory]	205
[SS:BP+SI+Memory]	206
[SS:BP+DI+Memory]	207
[SS:SI+Memory]	208
[SS:DI+Memory]	209
[SS:BP+Memory]	210
[SS:BX+Memory]	211
[SS:BX+SI+Value]	212
[SS:BX+DI+Value]	213
[SS:BP+SI+Value]	214
[SS:BP+DI+Value]	215
[SS:SI+Value]	216
[SS:DI+Value]	217
[SS:BP+Value]	218
[SS:BX+Value]	219
[DS:BX+SI]	220
[DS:BX+DI]	221
[DS:BP+SI]	222
[DS:BP+DI]	223
[DS:BX+SI+Memory]	224
[DS:BX+DI+Memory]	225
[DS:BP+SI+Memory]	226
[DS:BP+DI+Memory]	227
[DS:SI+Memory]	228
[DS:DI+Memory]	229
[DS:BP+Memory]	230
[DS:BX+Memory]	231
[DS:BX+SI+Value]	232
[DS:BX+DI+Value]	233
[DS:BP+SI+Value]	234
[DS:BP+DI+Value]	235
[DS:SI+Value]	236
[DS:DI+Value]	237
[DS:BP+Value]	238
[DS:BX+Value]	239

```
[FS:BX+SI]              240
[FS:BX+DI]              241
[FS:BP+SI]              242
[FS:BP+DI]              243
[FS:BX+SI+Memory]       244
[FS:BX+DI+Memory]       245
[FS:BP+SI+Memory]       246
[FS:BP+DI+Memory]       247
[FS:SI+Memory]          248
[FS:DI+Memory]          249
[FS:BP+Memory]          250
[FS:BX+Memory]          251
[FS:BX+SI+Value]        252
[FS:BX+DI+Value]        253
[FS:BP+SI+Value]        254
[FS:BP+DI+Value]        255
[FS:SI+Value]           256
[FS:DI+Value]           257
[FS:BP+Value]           258
[FS:BX+Value]           259

[GS:BX+SI]              260
[GS:BX+DI]              261
[GS:BP+SI]              262
[GS:BP+DI]              263
[GS:BX+SI+Memory]       264
[GS:BX+DI+Memory]       265
[GS:BP+SI+Memory]       266
[GS:BP+DI+Memory]       267
[GS:SI+Memory]          268
[GS:DI+Memory]          269
[GS:BP+Memory]          270
[GS:BX+Memory]          271
[GS:BX+SI+Value]        272
[GS:BX+DI+Value]        273
[GS:BP+SI+Value]        274
[GS:BP+DI+Value]        275
[GS:SI+Value]           276
[GS:DI+Value]           277
[GS:BP+Value]           278
[GS:BX+Value]           279

Value "7"               280
seg:offset (0000:0000)  300
```

Special rules for the "seg:offset" parameter:

---

* The "seg" value can be written with **DATA[R]** Immediate (parameter 2)
* The "offset" value can be written with **DATA[Q]** Immediate (parameter 1)
* You can **NOT** use other parameters with "seg:offset"

Data

The data tells the instruction what to do. See the list below:

DATA STRING = 123456789ABCDEFGHIJKLMNOPQRSTUVW

Pos.	Description
1-2	Instruction prefix ("--" if not used)
3	processor compability 80**x**86 (**x** = 0-6)
4	Parameter 1 type (**0 =** 8-bit, **1 =** 16-bit, **2 =** 32-bit, **3** = 64-bit, **4** = 48-bit, **5** = 80-bit)
5	Parameter 2 type (**0 =** 8-bit, **1 =** 16-bit, **2 =** 32-bit, **3** = 64-bit, **4** = 48-bit, **5** = 80-bit)
6	Use prefix 67h if segment is 16-bit?
7	Use prefix 66h if segment is 16-bit?
8	Use prefix 67h if segment is 32-bit?
9	Use prefix 66h if segment is 32-bit?
A-B	Segment overide ("--" if not used)
C-D	OpCode (1) ("--" if not used)
E-F	OpCode (2) ("--" if not used)
G	ModRM type **0 =** not used **1 =** standard reg/mem, reg/mem **2 =** digit reg/mem, digit (0-7) (get register/memory from *parameter 1*) **3 =** digit reg/mem, digit (0-7) (get register/memory from *parameter 2*) **4 =** special reg/mem, value (0-255) (get register/memory from *parameter 1*) **5 =** special reg/mem, value (0-255) (get register/memory from *parameter 2*)
H-I	ModRM value ("--" if not used)
J-K	Reserved byte (SIB) ("--" if not used)
L	Add Relocation address
M	Relative jump *(parameter 1)*
N	Relative jump *(parameter 2)*
O	Displacement *(parameter 1)*
P	Displacement *(parameter 2)*
Q	Immediate *(parameter 1)*
R	Immediate *(parameter 2)*
S	Get segment address (@data, @bss, ...) **(16-bit)** **0 =** not used **1 =** @CODE **2 =** @DATA **3 =** @SSEG **4 =** @BSS **5 =** @FSEG **6 =** @GSEG
T	Get segment address (@data, @bss, ...) **(32-bit)** **0 =** not used **1 =** @CODE **2 =** @DATA **3 =** @SSEG **4 =** @BSS **5 =** @FSEG **6 =** @GSEG
U	Add Register Code (0-7) to OpCode *(from Parameter 1)*
V	Add Register Code (0-7) to OpCode *(from Parameter 2)*
W	Use Paragraphs when get segment address ([S] or [T]) **0 =** No **1 =** Yes