

SETcc—Set Byte on Condition

Opcode	Instruction	Op/En	64-Bit Mode	Compat/Leg Mode	Description
OF 97	SETA <i>r/m8</i>	M	Valid	Valid	Set byte if above (CF=0 and ZF=0).
REX + OF 97	SETA <i>r/m8</i> *	M	Valid	N.E.	Set byte if above (CF=0 and ZF=0).
OF 93	SETAE <i>r/m8</i>	M	Valid	Valid	Set byte if above or equal (CF=0).
REX + OF 93	SETAE <i>r/m8</i> *	M	Valid	N.E.	Set byte if above or equal (CF=0).
OF 92	SETB <i>r/m8</i>	M	Valid	Valid	Set byte if below (CF=1).
REX + OF 92	SETB <i>r/m8</i> *	M	Valid	N.E.	Set byte if below (CF=1).
OF 96	SETBE <i>r/m8</i>	M	Valid	Valid	Set byte if below or equal (CF=1 or ZF=1).
REX + OF 96	SETBE <i>r/m8</i> *	M	Valid	N.E.	Set byte if below or equal (CF=1 or ZF=1).
OF 92	SETC <i>r/m8</i>	M	Valid	Valid	Set byte if carry (CF=1).
REX + OF 92	SETC <i>r/m8</i> *	M	Valid	N.E.	Set byte if carry (CF=1).
OF 94	SETE <i>r/m8</i>	M	Valid	Valid	Set byte if equal (ZF=1).
REX + OF 94	SETE <i>r/m8</i> *	M	Valid	N.E.	Set byte if equal (ZF=1).
OF 9F	SETG <i>r/m8</i>	M	Valid	Valid	Set byte if greater (ZF=0 and SF=OF).
REX + OF 9F	SETG <i>r/m8</i> *	M	Valid	N.E.	Set byte if greater (ZF=0 and SF=OF).
OF 9D	SETGE <i>r/m8</i>	M	Valid	Valid	Set byte if greater or equal (SF=OF).
REX + OF 9D	SETGE <i>r/m8</i> *	M	Valid	N.E.	Set byte if greater or equal (SF=OF).
OF 9C	SETL <i>r/m8</i>	M	Valid	Valid	Set byte if less (SF≠OF).
REX + OF 9C	SETL <i>r/m8</i> *	M	Valid	N.E.	Set byte if less (SF≠OF).
OF 9E	SETLE <i>r/m8</i>	M	Valid	Valid	Set byte if less or equal (ZF=1 or SF≠OF).
REX + OF 9E	SETLE <i>r/m8</i> *	M	Valid	N.E.	Set byte if less or equal (ZF=1 or SF≠OF).
OF 96	SETNA <i>r/m8</i>	M	Valid	Valid	Set byte if not above (CF=1 or ZF=1).

Opcode	Instruction	Op/ En	64-Bit Mode	Compat/ Leg Mode	Description
REX + 0F 96	SETNA <i>r/m8</i> *	M	Valid	N.E.	Set byte if not above (CF=1 or ZF=1).
0F 92	SETNAE <i>r/m8</i>	M	Valid	Valid	Set byte if not above or equal (CF=1).
REX + 0F 92	SETNAE <i>r/m8</i> *	M	Valid	N.E.	Set byte if not above or equal (CF=1).
0F 93	SETNB <i>r/m8</i>	M	Valid	Valid	Set byte if not below (CF=0).
REX + 0F 93	SETNB <i>r/m8</i> *	M	Valid	N.E.	Set byte if not below (CF=0).
0F 97	SETNBE <i>r/m8</i>	M	Valid	Valid	Set byte if not below or equal (CF=0 and ZF=0).
REX + 0F 97	SETNBE <i>r/m8</i> *	M	Valid	N.E.	Set byte if not below or equal (CF=0 and ZF=0).
0F 93	SETNC <i>r/m8</i>	M	Valid	Valid	Set byte if not carry (CF=0).
REX + 0F 93	SETNC <i>r/m8</i> *	M	Valid	N.E.	Set byte if not carry (CF=0).
0F 95	SETNE <i>r/m8</i>	M	Valid	Valid	Set byte if not equal (ZF=0).
REX + 0F 95	SETNE <i>r/m8</i> *	M	Valid	N.E.	Set byte if not equal (ZF=0).
0F 9E	SETNG <i>r/m8</i>	M	Valid	Valid	Set byte if not greater (ZF=1 or SF≠ OF)
REX + 0F 9E	SETNG <i>r/m8</i> *	M	Valid	N.E.	Set byte if not greater (ZF=1 or SF≠ OF).
0F 9C	SETNGE <i>r/m8</i>	M	Valid	Valid	Set byte if not greater or equal (SF≠ OF).
REX + 0F 9C	SETNGE <i>r/m8</i> *	M	Valid	N.E.	Set byte if not greater or equal (SF≠ OF).
0F 9D	SETNL <i>r/m8</i>	M	Valid	Valid	Set byte if not less (SF=OF).
REX + 0F 9D	SETNL <i>r/m8</i> *	M	Valid	N.E.	Set byte if not less (SF=OF).
0F 9F	SETNLE <i>r/m8</i>	M	Valid	Valid	Set byte if not less or equal (ZF=0 and SF=OF).
REX + 0F 9F	SETNLE <i>r/m8</i> *	M	Valid	N.E.	Set byte if not less or equal (ZF=0 and SF=OF).
0F 91	SETNO <i>r/m8</i>	M	Valid	Valid	Set byte if not overflow (OF=0).
REX + 0F 91	SETNO <i>r/m8</i> *	M	Valid	N.E.	Set byte if not overflow (OF=0).
0F 9B	SETNP <i>r/m8</i>	M	Valid	Valid	Set byte if not parity (PF=0).

Opcode	Instruction	Op/En	64-Bit Mode	Compat/Leg Mode	Description
REX + 0F 9B	SETNP <i>r/m8</i> *	M	Valid	N.E.	Set byte if not parity (PF=0).
0F 99	SETNS <i>r/m8</i>	M	Valid	Valid	Set byte if not sign (SF=0).
REX + 0F 99	SETNS <i>r/m8</i> *	M	Valid	N.E.	Set byte if not sign (SF=0).
0F 95	SETNZ <i>r/m8</i>	M	Valid	Valid	Set byte if not zero (ZF=0).
REX + 0F 95	SETNZ <i>r/m8</i> *	M	Valid	N.E.	Set byte if not zero (ZF=0).
0F 90	SETO <i>r/m8</i>	M	Valid	Valid	Set byte if overflow (OF=1)
REX + 0F 90	SETO <i>r/m8</i> *	M	Valid	N.E.	Set byte if overflow (OF=1).
0F 9A	SETP <i>r/m8</i>	M	Valid	Valid	Set byte if parity (PF=1).
REX + 0F 9A	SETP <i>r/m8</i> *	M	Valid	N.E.	Set byte if parity (PF=1).
0F 9A	SETPE <i>r/m8</i>	M	Valid	Valid	Set byte if parity even (PF=1).
REX + 0F 9A	SETPE <i>r/m8</i> *	M	Valid	N.E.	Set byte if parity even (PF=1).
0F 9B	SETPO <i>r/m8</i>	M	Valid	Valid	Set byte if parity odd (PF=0).
REX + 0F 9B	SETPO <i>r/m8</i> *	M	Valid	N.E.	Set byte if parity odd (PF=0).
0F 98	SETS <i>r/m8</i>	M	Valid	Valid	Set byte if sign (SF=1).
REX + 0F 98	SETS <i>r/m8</i> *	M	Valid	N.E.	Set byte if sign (SF=1).
0F 94	SETZ <i>r/m8</i>	M	Valid	Valid	Set byte if zero (ZF=1).
REX + 0F 94	SETZ <i>r/m8</i> *	M	Valid	N.E.	Set byte if zero (ZF=1).

NOTES:

* In 64-bit mode, *r/m8* can not be encoded to access the following byte registers if a REX prefix is used: AH, BH, CH, DH.

Instruction Operand Encoding

Op/En	Operand 1	Operand 2	Operand 3	Operand 4
M	ModRM:r/m (r)	NA	NA	NA

Description

Sets the destination operand to 0 or 1 depending on the settings of the status flags (CF, SF, OF, ZF, and PF) in the EFLAGS register. The destination operand points to a byte register or a byte in memory. The condition code suffix (*cc*) indicates the condition being tested for.

The terms “above” and “below” are associated with the CF flag and refer to the relationship between two unsigned integer values. The terms “greater” and “less” are associated with the SF and OF flags and refer to the relationship between two signed integer values.

Many of the SETcc instruction opcodes have alternate mnemonics. For example, SETG (set byte if greater) and SETNLE (set if not less or equal) have the same opcode and test for the same condition: ZF equals 0 and SF equals OF. These alternate mnemonics are provided to make code more intelligible. Appendix B, “EFLAGS Condition Codes,” in the *Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 1*, shows the alternate mnemonics for various test conditions.

Some languages represent a logical one as an integer with all bits set. This representation can be obtained by choosing the logically opposite condition for the SETcc instruction, then decrementing the result. For example, to test for overflow, use the SETNO instruction, then decrement the result.

In IA-64 mode, the operand size is fixed at 8 bits. Use of REX prefix enable uniform addressing to additional byte registers. Otherwise, this instruction’s operation is the same as in legacy mode and compatibility mode.

Operation

```
IF condition
    THEN DEST ← 1;
    ELSE DEST ← 0;
FI;
```

Flags Affected

None.

Protected Mode Exceptions

#GP(0)	If the destination is located in a non-writable segment. If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit. If the DS, ES, FS, or GS register contains a NULL segment selector.
#SS(0)	If a memory operand effective address is outside the SS segment limit.
#PF(fault-code)	If a page fault occurs.
#UD	If the LOCK prefix is used.

Real-Address Mode Exceptions

#GP	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
-----	---

INSTRUCTION SET REFERENCE, M-Z

#SS	If a memory operand effective address is outside the SS segment limit.
#UD	If the LOCK prefix is used.

Virtual-8086 Mode Exceptions

#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
#SS(0)	If a memory operand effective address is outside the SS segment limit.
#PF(fault-code)	If a page fault occurs.
#UD	If the LOCK prefix is used.

Compatibility Mode Exceptions

Same exceptions as in protected mode.

64-Bit Mode Exceptions

#SS(0)	If a memory address referencing the SS segment is in a non-canonical form.
#GP(0)	If the memory address is in a non-canonical form.
#PF(fault-code)	If a page fault occurs.
#UD	If the LOCK prefix is used.