Optimizer Options

When the Natural Optimizer has been activated, you can specify checks by setting the options explained in this section.

The options cannot be used for specifying statements to be optimized.

This chapter covers the following topics:

- List of Options

- PGEN Option

- Influence of other Natural Parameters

List of Options

The following table lists and describes NOC options. Default values are underlined (this is the value that will be assumed if the option is not present).

A NOC option consists of a string surrounded by brackets or single quotation marks (except in the Natural `OPTIONS` statement), with options separated by commas. Some options have values, while the very existence of some options in the option string is sufficient to modify the environment.

The following rules apply:

- Optional clauses are surrounded by square brackets [ ].

- Choices are surrounded by curly braces { }.

- Each choice is separated by vertical lines "|".

- Only one of these choices can be specified;

  `ON` is equivalent to `Y` (Yes),

  `OFF` to `N` (No).

- Options specified without the optional clause `ON` or `OFF` (if applicable), or their equivalent values, are interpreted as set to `ON`. For example, `OVFLW` is identical to `OVFLW=ON`.

- Except for the option `OFF`, any specified option switches on optimizing (as if `ON` was specified) and the default values apply. For example, `INDEX` is identical to `ON,INDEX`.

Option	Explanation
ABEND	Forces the Natural Optimizer Compiler to generate code which causes Natural to be abnormally terminated immediately when the ABEND option is encountered by the Natural Optimizer Compiler during compilation. The option must appear by itself or it will be ignored. Other parameters are not changed or reset by this option. This option can be useful for debugging purposes.
CACHE[={ON| OFF |Y|N}]	Switches variable caching on or off. See also *Variable Caching* in the section Performance Considerations.
CPU= /370	Specifies the target architecture. The /370 option is valid for IBM and Siemens.
DIGTCHCK[={ON| OFF |Y|N}]	Specifies whether the digits of packed and unpacked numeric fields (formats P and N) are to be checked when moving to another variable of the same type and precision. For example, if DIGTCHCK is ON and an unpacked numeric variable (format N) contains an invalid digit, such as X'FA', moving to another unpacked numeric variable with the same precision will generate a S0C7 (or NAT0954) error. If DIGTCHCK is OFF, no error is generated but the generated code is much faster.
ERRDUMP[={ON| OFF |Y|N}]	Specifies whether NOC should abend if an error condition is detected during the compile phase. This is useful for debugging the Natural Optimizer Compiler itself.
INDEX[={ON| OFF |Y|N}]	Specifies whether array indexes will be checked for out-of-bound values in the optimized code. See also the *Warning* below.
INDX[={ON| OFF |Y|N}]	Specifies whether array indexes will be checked for out-of-bound values in the optimized code. Additionally, RANGE will be set on. Therefore, this option is equivalent to INDEX=ON,RANGE=ON. See also the *Warning* below.
IO[={ON| OFF|Y|N}]	Reserved for future use.
LOOPS[={ON| OFF |Y|N}]	Provided for compatibility reasons only. No effect.
MIX[={ON| OFF |Y|N}]	Provided for compatibility reasons only. No effect.
NODBG[={ON| OFF|Y|N}]	If you have not set NODBG=OFF/N (default), the Natural Debugger can not be used to debug optimized code, because the Natural Debugger might not receive control for optimized statements. No additional code will be generated, the program will not run slower and will not consume more CPU time. If you have not set NODBG=ON/Y, additional code is generated to check whether TEST mode has been set on and the functionality of the Natural Debugger will not be limited. See also NODBG in the section *Performance Considerations*.

Option	Explanation
NOSGNTR[={ON| OFF |Y|N}]	Applies to packed numbers only. If NOSGNTR=OFF (default), signs of positive packed numbers which are the result of an arithmetic operation or the target of an assignment are set according to the COMPOPT parameter PSIGNF. If NOSGNTR=ON, the signs resulting from execution of the generated machine instruction are left unchanged. See also the section Influence of other Natural Parameters.
ON	Switches on optimizing. If no additional option is specified, the default value defined for each option is in effect. As indicated in the Warning below, this may cause unintended results, in particular regarding the options INDEX, INDX, OVFLW, and RANGE.
OFF	Switches off optimizing.
OPTLEV={ 2|3}	Specifies optimization level - roughly equivalent to the number of passes through the program. OPTLEV=3 is useful when PGEN is specified, since some branch targets cannot be determined during the first pass and PGEN output is made during the last pass. Thus, some values may be shown improperly.
OVFLW[={ON| OFF |Y|N}]	Specifies whether checks for overflow in arithmetic operations or assignments will be included in the optimized code. See also the *Warning* below.
PGEN[={ON| OFF |Y|N}]	Specifies whether a disassembly of the optimized code should be output. This option also enables all other tracing options. See also *PGEN Option* below.
RANGE[={ON| OFF |Y|N}]	Specifies whether range checks will be performed in operations with arrays. This ensures that array ranges will have an equal number of elements in corresponding dimensions of all operands. See also the *Warning* below.
SIGNCHCK[={ ON|OFF|Y|N}]	Specifies whether the result of a multiplication with a packed or unpacked numeric multiplier should be checked for a negative zero. If zero is multiplied by a negative number, the MP machine instruction generates a negative zero result. If SIGNCHCK is on, this negative zero is converted to a positive zero. The check for a negative zero is done for every multiplication with a packed or unpacked numeric multiplier.
TRENTRY	For internal use by Software AG only. Do not change the setting of this parameter.

Option	Explanation
ZD[={ <u>ON</u>|OFF|Y|N}]	Specifies whether divisors should be checked for zero. If this option is specified, then code is inserted, so that the program behaves according to the ZD profile parameter of Natural, that is, Natural error NAT1302 is issued or the result is zero. If this option is not specified, Natural error NAT0954 occurs if the divisor is zero. See also *ZD - Zero-Division Check* in the Natural *Parameter Reference* documentation.

⚠ **Warning:**
For INDEX, INDX, OVFLW, and RANGE:
Apply values OFF and N with care. Suppressing overflow checking or array index checking may allow incorrect programs to lead to unpredictable results, storage corruption, or abnormal terminating. See also the *Example of INDEX and OVFLW* below which demonstrates the impact of INDEX and OVFLW.

Below is information on:

- Example of INDEX and OVFLW

- Optimum Code Generation

Example of INDEX and OVFLW

```
DEFINE DATA LOCAL
...
1 P1 (P1/9)
...
1 P3 (P3/9)
...
1 I (I4)
1 J (I4)
1 K (I4)
1 L (I4)
END-DEFINE
...
P1(I:J) := P3(K:L)
...
END
```

Explanation of Example

With INDX=ON or INDEX=ON set, code is generated to verify that I, J, K and L are within the ranges defined for P1 and P3 respectively.

With INDX=ON or RANGE=ON set, code is generated to verify that I:J and K:L denote ranges of the same length.

With `OVFLW=ON` set, code is generated to verify that the value of `P3` fits into the corresponding `P1` variable.

For example: Value `100` would cause an overflow here.

Example Error Situation:

If one of the occurrences of `P3` contains the value `100`, with `OVFLW=OFF` set, the value assigned to the corresponding `P1` occurrence will be zero. If the index variable `I` is zero or greater than `9`, with `INDX=OFF` set, storage areas that do not belong to Array `P1` will be corrupted. If these options (`OVFLW` and `INDX`) are set to `ON`, a Natural error occurs like it does in standard Natural runtime.

For the NOC option specified above, additional code is generated. However, this is well compensated for by the advantage of a check that, for example, protects against hard-to-debug errors. Undetected errors can, of course, lead to unpredictable results.

Optimum Code Generation

To assure that the least amount of code is generated and thus achieve optimum performance, use:

```
OPT='NODBG,NOSGNTR,SIGNCHCK=OFF,ZD=OFF'
```

However, only apply this setting to programming objects that have been thoroughly debugged; see also the *Warning*.

PGEN Option

The `PGEN` option causes the Natural Optimizer Compiler to output the generated code and internal Natural structures. Thus, code and structures can be examined, for example, for bug fixing, performance review and support issues.

An understanding of IBM's /370 assembler is required to interpret the results produced by the `PGEN` option.

We recommend that you use this option with the assistance of your local Software AG representative.

Below is information on:

- Setting PGEN

- Sub-Options of the PGEN Option

- Output of the PGEN Option

- Working with the PGEN Output

Setting PGEN

To use the `PGEN` facility, set the `PGEN` option when activating on the Optimizer Compiler.

Since the buffer is kept in memory, it is possible that the user thread will not be big enough to hold the trace information. In this case, try setting `PGEN` on only for the portion of the program which is to be traced, for example:

`OPTIONS MCG=(PGEN=ON,TRGPT=ON)`
or
`OPTIONS MCG=+PGEN,TRGPT`
	Turns tracing on, including tracing of the GPT entries

`OPTIONS MCG=(PGEN=OFF)`
or
`OPTIONS MCG=-PGEN`
	Turns tracing off

Various options affect the content of the output. The basic `PGEN` option causes a formatted listing of Natural source lines and a disassembly of the corresponding code to be generated and kept in memory for extraction by the `NOCSHOW` utility as described below, under *Output of the PGEN Option*.

The `TRSTMT`, `TRGPT`, `TRMPT` and `TRVDT` options cause hex dumps of internal data structures associated with each line to be output.

The `TRBASES` and `TRCACHE` options cause information on base registers and cache variables to be printed out.

Sub-Options of the PGEN Option

The following table describes the options when `PGEN=ON`. For an explanation of the syntax used see the introduction to *List of Options* above.

Option	Explanation
`LPP={5|..| 55 |..|255}`	Lines-per-page for the trace output, only used when `TREXT=ON`.
`NOsrcE[={ON| OFF |Y|N}]`	If `NOsrcE=OFF`, the Natural source statement is included in the output.
`TRACELEV={ 0 |..|255}`	Specifies the trace level. Each bit in this one byte value specifies a buffer type to trace; these bits can be set on by using the `TRxxx` options as well.
`TRBASES[={ON| OFF |Y|N}]`	Specifies whether base register allocations are traced.
`TRCACHE[={ON| OFF|Y|N}]`	Specifies whether `CACHE` entries are traced.
`TREXT[={ON| OFF |Y|N}]`	If `TREXT=ON`, trace is directed to the user exit `NOCPRINT` as described below.
`TRGPT[={ON| OFF |Y|N}]`	Specifies whether `GPT` entries are traced.
`TRMPT[=ON| OFF |Y|N}]`	Specifies whether `MPT` entries are traced.
`TRSTMT[={ON| OFF |Y|N}]`	Specifies whether `STMT` entries are traced.
`TRVDT[={ON| OFF |Y|N}]`	Specifies whether `VDT` entries are traced.

See also the examples below.

Output of the PGEN Option

There are two places to where the Natural Optimizer Compiler can direct the output of PGEN:

- internal Buffer
- User Exit NOCPRINT

internal Buffer

The contents of this buffer is overwritten each time a CHECK, CAT, STOW or RUN command is executed. A system utility NOCSHOW is provided whereby the contents of this buffer can be viewed, searched or printed.

▶ To invoke the NOCSHOW utility

- Enter the direct command NOCSHOW after a CHECK, STOW, CAT or RUN where the Natural Optimizer Compiler has been active.

The following PF keys are available on the screen:

Key	Function
PF2	Position to top of output
PF4	Position one line backward
PF5	Position one line forward
PF6	Print to Natural printer support No.1
PF7	Position one page backward
PF8	Position one page forward
PF10	Scan for text string
PF11	Repeat scan

User Exit NOCPRINT

If TREXT=ON is specified, the Natural Optimizer Compiler passes every output line to the user exit NOCPRINT instead of adding it to the trace buffer.

NOCPRINT is invoked following normal OS register conventions. Register 1 points to a full word containing the address of the 81 byte print line with ANSI carriage control characters in position 1. Register 13 points to an area of 18*4 bytes which may be used as a save area. Register 14 contains the return address and Register 15 contains the entry address of NOCPRINT.

The user exit NOCPRINT can be written in any language which supports the register conventions described above. It must be linked to the Natural nucleus together with the Natural Optimizer Compiler nucleus.

Working with the PGEN Output

This section provides hints and explanations on how to interpret the output created with the PGEN option.

- At the top of the PGEN output are some disassembled lines which do not appear to belong to any source line. These are the instructions which make up the prologue, which is executed whenever control passes from non-optimized to optimized code. Permanent base registers are loaded and control is passed to the correct point in the prologue. See *Example Section A* below.

- Sometimes a lot of source lines are printed without any code. This is because the Natural compiler puts a single line number in the object of statements which may span more than one line. See *Example Section B* below.

- If the NODBG=OFF (default) has been specified, a sequence of instructions is generated at the start of each Natural statement:

```
BALR R9,R11
DC X'....'
```

  This sequence sets the line number (in case of error) and checks whether the TEST mode is switched ON. Without this sequence, debugging of NOC-compiled statements by the Natural Debugger is not possible. See *Example Section C* below.

- Sometimes there is a line break between disassembled lines. This break indicates an internal statement separation. It happens because often a single Natural statement will generate multiple internal (pseudo-code) statements.

Example Section A:

```
000000 5880 D354          L     R8,RTADR+4
000004 5870 D370          L     R7,RTADR+32
000008 4810 6006          LH    R1,6(,R6)
00000C 1F60               SLR   R6,R0
00000E 47F1 A000          BC    15,0(R1,R10)
```

Example Section B:

```
0010 OPTIONS MCG=(PGEN=ON,TRGPT=ON)
0020 DEFINE DATA LOCAL
0030 1 I(I4)
0040 1 P(P7.2)
0050 1 T(P7.2)
0060 END-DEFINE
0070 *

0080 SETTIME
0090 *
```

```
000012 45E0 B040          BAL   R14,RETH
000016 0036               DC    X'0036'
```

```
0100 FOR I=1 TO 100000
```

Example Section C:

```
000018 059B                 BALR  R9,R11
   00001A 003E                  DC   X'003E'
   00001C D203 7000 833B        MVC   I,#VAR033B

   000022 059B                  BALR  R9,R11
   000024 004C                  DC   X'004C'
   000026 47F0 A040             BC   15,64(,R10)

   00002A 059B                  BALR  R9,R11
   00002C 005A                  DC   X'005A'
   00002E BFFF 8343             ICM   R15,15,#VAR0343
   000032 BF0F 7000             ICM   R0,15,I
   000036 1A0F                  AR   R0,R15
   000038 BE0F 7000             STCM  R0,15,I

   00003C 059B                  BALR  R9,R11
   00003E 006C                  DC   X'006C'
   000040 BFFF 833F             ICM   R15,15,#VAR033F
   000044 BF0F 7000             ICM   R0,15,I
   000048 190F                  CR   R0,R15
   00004A 4720 A066             BC   2,102(,R10)

   0110   ADD 1.00 TO P

   00004E 059B                  BALR  R9,R11
   000050 0082                  DC   X'0082'
   000052 FA41 7004 8347        AP   P,#VAR0347
   000058 DC00 7008 B488        TR   P+4(1),PSGNTR

   0120 END-FOR
   0130 *

   00005E 059B                  BALR  R9,R11
   000060 0094                  DC   X'0094'
   000062 47F0 A02A             BC   15,42(,R10)

   0140 T:=*TIMD(0080)

   000066 059B                  BALR  R9,R11
   000068 009C                  DC   X'009C'
   00006A 45E0 B0D8             BAL   R14,SYSFUNC
   00006E 0330 B881             DC   X'0330B881'

   000072 F246 7009 8330        PACK  T,#VAR0330
   000078 F040 7009 0002        SRP   T,2,0
   00007E DC00 700D B488        TR   T+4(1),PSGNTR

   0150 T:=T / 10
   0160 *

   000084 059B                  BALR  R9,R11
   000086 00AE                  DC   X'00AE'
   000088 F864 D100 7009        ZAP   OP1(7),T
   00008E F811 D130 8349        ZAP   WORK2(2),#VAR0349
   000094 45E0 B104             BAL   R14,ZDCHECK
   000098 F240 7009 B355        PACK  T,ZEROZ
   00009E 47F0 E01C             BC   15,28(,R14)
   0000A2 FD61 D100 8349        DP   OP1(7),#VAR0349
   0000A8 D204 7009 D100        MVC   T,OP1
   0000AE DC00 700D B488        TR   T+4(1),PSGNTR
```

```
   0170 DISPLAY 'ELAPSED TIME (S)' T

 0000B4 45E0 B040              BAL    R14,RETH
 0000B8 00C0                   DC     X'00C0'

   0180 END
```

Influence of other Natural Parameters

The global parameter ZD influences the behavior of the NOC compiler. See the description of the ZD option as described under *List of Options* above.

The COMPOPT parameter PSIGNF (see also the system command COMPOPT in the Natural *System Commands* documentation) influences the behavior by forcing the signs of positive packed decimal numbers to F if ON, and to C if OFF. The parameter is applied if NOSGNTR=OFF is specified.

See the chart below for packed data (Format P) ":"

NOSGNTR=OFF	and	PSIGNF=ON	All signs are normalized to F (default).
NOSGNTR=OFF	and	PSIGNF=OFF	All signs are normalized to C.
NOSGNTR=ON			All signs are left as they were generated by the last operation.

For numeric data (Format N) the signs are always normalized to F, regardless of the settings of NOSGNTR and PSIGNF.