

# Cross Reference Listing

Cross reference listing of MISRA-C:2004 Rules against messages generated

copyright: © 2007 Programming Research

## **MISRA-C:2004 Rule 1.1** (Required)

All code shall conform to ISO 9899:1990 C programming language, ISO 9899, amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996

QAC Messages: 0180, 0240, 0241, 0246, 0410, 0551, 0601, 0604, 0609, 0611, 0612, 0614, 0639, 0647, 0662, 0715, 0739, 0810, 0828, 0830, 0857, 0858, 0859, 0875, 0930, 1003, 1006, 1008, 1014, 1015, 1018, 1019, 1020, 1021, 1022, 1026, 1027, 1028, 1029, 1030, 1031, 3664

## **MISRA-C:2004 Rule 1.2** (Required)

No reliance shall be placed on undefined or unspecified behaviour.

QAC Messages: 0160, 0161, 0162, 0163, 0164, 0165, 0166, 0167, 0168, 0169, 0170, 0171, 0172, 0173, 0174, 0175, 0176, 0177, 0178, 0179, 0184, 0185, 0186, 0190, 0191, 0192, 0193, 0194, 0195, 0196, 0197, 0198, 0199, 0200, 0201, 0203, 0204, 0206, 0207, 0208, 0275, 0304, 0309, 0337, 0475, 0503, 0504, 0505, 0507, 0508, 0509, 0541, 0543, 0546, 0586, 0625, 0632, 0654, 0658, 0661, 0667, 0668, 0672, 0675, 0676, 0678, 0680, 0706, 0747, 0837, 0864, 0865, 0867, 0872, 0874, 3311, 3312, 3437, 3438, 3680

## **MISRA-C:2004 Rule 1.3** (Required)

Multiple compilers and/or languages shall only be used if there is a common defined interface standard for object code to which the languages/compiler/assemblers conform.

**Enforcement:** NSE

## **MISRA-C:2004 Rule 1.4** (Required)

The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.

**Enforcement:** NSE

## **MISRA-C:2004 Rule 1.5** (Advisory)

Floating-point implementations should comply with a defined floating-point standard.

**Enforcement:** NSE

## **MISRA-C:2004 Rule 2.1** (Required)

Assembly language shall be encapsulated and isolated.

QAC Messages: 3006

## **MISRA-C:2004 Rule 2.2** (Required)

Source code shall only use C-style comments.

QAC Messages: 1011

## **MISRA-C:2004 Rule 2.3** (Required)

The character sequence `/*` shall not be used within a comment.

QAC Messages: 3108

## **MISRA-C:2004 Rule 2.4** (Advisory)

Sections of code should not be 'commented out'.

**Enforcement:** NSE

**MISRA-C:2004 Rule 3.1 (Required)**

All usage of implementation-defined behaviour shall be documented.

QAC Messages: 0202, 0271, 0277, 0284, 0285, 0286, 0287, 0288, 0289, 0294, 0295, 0308, 0878, 3702, 3703, 3704, 3705, 3706, 3707, 3832, 3833, 3902, 3903, 3904, 3905, 3906, 3907, 4032, 4033

**MISRA-C:2004 Rule 3.2 (Required)**

The character set and the corresponding encoding shall be documented.

**Enforcement:** NSE

**MISRA-C:2004 Rule 3.3 (Advisory)**

The implementation of integer division in the chosen compiler should be determined, documented and taken into account.

**Enforcement:** NSE

**MISRA-C:2004 Rule 3.4 (Required)**

All uses of the #pragma directive shall be documented and explained.

QAC Messages: 3116

**MISRA-C:2004 Rule 3.5 (Required)**

If it is being relied upon, the implementation-defined behaviour and packing of bitfields shall be documented.

**Enforcement:** NSE

**MISRA-C:2004 Rule 3.6 (Required)**

All libraries used in production code shall be written to comply with the provisions of this document, and shall have been subject to appropriate validation.

**Enforcement:** NSE

**MISRA-C:2004 Rule 4.1 (Required)**

Only those escape sequences that are defined in the ISO C standard shall be used.

QAC Messages: 0235

**MISRA-C:2004 Rule 4.2 (Required)**

Trigraphs shall not be used.

QAC Messages: 3601

**MISRA-C:2004 Rule 5.1 (Required)**

Identifiers (internal and external) shall not rely on the significance of more than 31 characters.

QAC Messages: 0777, 0779

**MISRA-C:2004 Rule 5.2 (Required)**

Identifiers in an inner scope shall not use the same name as an identifier in an outer scope, and therefore hide that identifier.

QAC Messages: 2547, 3334

**MISRA-C:2004 Rule 5.3 (Required)**

A typedef name shall be a unique identifier.

QAC Messages: 1506, 1507, 1508, 3448

**MISRA-C:2004 Rule 5.4 (Required)**

A tag name shall be a unique identifier.

QAC Messages: 0547

**MISRA-C:2004 Rule 5.5 (Advisory)**

No object or function identifier with static storage duration should be reused.

QAC Messages: 1525, 1526, 1527, 1528, 1529

**MISRA-C:2004 Rule 5.6** (Advisory)

No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure and union member names.

QAC Messages: 0780, 0781

**MISRA-C:2004 Rule 5.7** (Advisory)

No identifier name should be reused.

**MISRA-C:2004 Rule 6.1** (Required)

The plain char type shall be used only for the storage and use of character values.

QAC Messages: 3711, 3722, 3733, 3744, 3755, 3766, 3777, 3788, 3850, 3863, 3911, 3922, 3933, 3944, 3955, 3966, 3977, 3988, 4050, 4063

**MISRA-C:2004 Rule 6.2** (Required)

Signed and unsigned char type shall be used only for the storage and use of numeric values.

QAC Messages: 3700, 3701, 3900, 3901

**MISRA-C:2004 Rule 6.3** (Advisory)

Typedefs that indicate size and signedness should be used in place of the basic types.

QAC Messages: 5013

**MISRA-C:2004 Rule 6.4** (Required)

Bit fields shall only be defined to be of type unsigned int or signed int.

QAC Messages: 0634, 0635

**MISRA-C:2004 Rule 6.5** (Required)

Bit fields of type signed int shall be at least 2 bits long.

QAC Messages: 3660

**MISRA-C:2004 Rule 7.1** (Required)

Octal constants (other than zero) and octal escape sequences shall not be used.

QAC Messages: 0336, 0339, 3628

**MISRA-C:2004 Rule 8.1** (Required)

Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call.

QAC Messages: 3002, 3335, 3450

**MISRA-C:2004 Rule 8.2** (Required)

Whenever an object or function is declared or defined, its type shall be explicitly stated.

QAC Messages: 2050, 2051

**MISRA-C:2004 Rule 8.3** (Required)

For each function parameter the type given in the declaration and definition shall be identical, and the return types shall also be identical.

QAC Messages: 1331, 1332, 1333, 3320

**MISRA-C:2004 Rule 8.4** (Required)

If objects or functions are declared more than once their types shall be compatible.

QAC Messages: 0626, 0627, 0628, 1510

**MISRA-C:2004 Rule 8.5 (Required)**

There shall be no definitions of objects or functions in a header file.

QAC Messages: 3406

**MISRA-C:2004 Rule 8.6 (Required)**

Functions shall be declared at file scope.

QAC Messages: 3221

**MISRA-C:2004 Rule 8.7 (Required)**

Objects shall be defined at block scope if they are only accessed from within a single function.

QAC Messages: 1514, 3218

**MISRA-C:2004 Rule 8.8 (Required)**

An external object or function shall be declared in one and only one file.

QAC Messages: 1513, 3222, 3408, 3447, 3451

**MISRA-C:2004 Rule 8.9 (Required)**

An identifier with external linkage shall have exactly one external definition.

QAC Messages: 0630, 1509

**MISRA-C:2004 Rule 8.10 (Required)**

All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.

QAC Messages: 1504, 1505

**MISRA-C:2004 Rule 8.11 (Required)**

The static storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage.

QAC Messages: 3224

**MISRA-C:2004 Rule 8.12 (Required)**

When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation.

QAC Messages: 3684

**MISRA-C:2004 Rule 9.1 (Required)**

All automatic variables shall have been assigned a value before being used.

QAC Messages: 3321, 3347, 3348, 3349, 3353, 3354

**MISRA-C:2004 Rule 9.2 (Required)**

Braces shall be used to indicate and match the structure in the non-zero initialisation of arrays and structures.

QAC Messages: 0679, 0686, 0693, 0694

**MISRA-C:2004 Rule 9.3 (Required)**

In an enumerator list, the '=' construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised.

QAC Messages: 0723

**MISRA-C:2004 Rule 10.1 (Required)**

The value of an expression of integer type shall not be implicitly converted to a different underlying type if: a) it is not a conversion to a wider integer type of the same signedness, or b) the expression is complex, or c) the expression is not constant and is a function argument, or d) the expression is not constant and is a return expression

QAC Messages: 0274, 1317, 1401, 1402, 1403, 1411, 1412, 1413, 1421, 1422, 1423, 1431, 1433, 1441, 1442, 1443, 3306, 3708, 3709, 3710, 3712, 3713, 3715, 3717, 3719, 3720, 3721, 3723, 3725, 3727, 3729, 3730, 3731, 3732, 3734, 3735, 3736, 3738, 3740, 3741, 3742, 3743, 3745, 3746, 3747, 3748, 3750, 3752, 3753, 3754, 3756, 3757, 3758, 3759, 3760, 3762, 3763, 3764, 3765, 3767, 3768, 3769, 3770, 3771, 3772, 3774, 3775, 3776, 3778, 3779, 3780, 3781, 3782, 3783, 3784, 3785, 3786, 3787, 3789, 3790, 3791, 3792, 3793, 3794, 3795, 3796, 3797, 3798, 3834, 3837, 3839, 3840, 3843, 3844, 3847, 3848, 3851, 3852, 3853, 3854, 3855, 3856, 3857, 3858, 3859, 3860, 3861, 3862, 3864, 3865, 3866, 3867, 3868, 3869, 3870, 3871, 3872, 3873, 3874, 3875, 3912, 3913, 3915, 3917, 3919, 3920, 3921, 3923, 3925, 3927, 3929, 3930, 3931, 3932, 3934, 3935, 3936, 3938, 3940, 3941, 3942, 3943, 3945, 3946, 3947, 3948, 3950, 3952, 3953, 3954, 3956, 3957, 3958, 3959, 3960, 3962, 3963, 3964, 3965, 3967, 3968, 3969, 3970, 3971, 3972, 3974, 3975, 3976, 3978, 3979, 3980, 3981, 3982, 3983, 3984, 3985, 3986, 3987, 3989, 3990, 3991, 3992, 3993, 3994, 3995, 3996, 3997, 3998, 4034, 4037, 4039, 4040, 4043, 4044, 4047, 4048, 4051, 4052, 4053, 4054, 4055, 4056, 4057, 4058, 4059, 4060, 4061, 4062, 4064, 4065, 4066, 4067, 4068, 4069, 4070, 4071, 4072, 4073, 4074, 4075, 4120

**MISRA-C:2004 Rule 10.2 (Required)**

The value of an expression of floating type shall not be implicitly converted to a different type if: a) it is not a conversion to a wider floating type, or b) the expression is complex, or c) the expression is a function argument, or d) the expression is a return expression

QAC Messages: 3799, 3800, 3801, 3802, 3803, 3804, 3805, 3806, 3807, 3810, 3811, 3812, 3813, 3814, 3815, 3816, 3817, 3818, 3819, 3821, 3822, 3823, 3824, 3825, 3826, 3827, 3828, 3829, 3830, 3831, 3876, 3877, 3878, 3879, 3880, 3881, 3999, 4000, 4001, 4002, 4003, 4004, 4005, 4006, 4007, 4010, 4011, 4012, 4013, 4014, 4015, 4016, 4017, 4018, 4019, 4021, 4022, 4023, 4024, 4025, 4026, 4027, 4028, 4029, 4030, 4031, 4076, 4077, 4078, 4079, 4080, 4081, 4123, 4124, 4125

**MISRA-C:2004 Rule 10.3 (Required)**

The value of a complex expression of integer type may only be cast to a type that is narrower and of the same signedness as the underlying type of the expression.

QAC Messages: 4121

**MISRA-C:2004 Rule 10.4 (Required)**

The value of a complex expression of floating type may only be cast to a narrower floating type.

QAC Messages: 4126, 4127, 4128

**MISRA-C:2004 Rule 10.5 (Required)**

If the bitwise operators ~ and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand.

**MISRA-C:2004 Rule 10.6 (Required)**

A "U" suffix shall be applied to all constants of unsigned type.

QAC Messages: 1281

**MISRA-C:2004 Rule 11.1 (Required)**

Conversions shall not be performed between a pointer to a function and any type other than an integral type.

QAC Messages: 0302, 0307, 0313

**MISRA-C:2004 Rule 11.2 (Required)**

Conversions shall not be performed between a pointer to object and any type other than an integral type, another pointer to object type or a pointer to void.

QAC Messages: 0301

**MISRA-C:2004 Rule 11.3 (Advisory)**

A cast should not be performed between a pointer type and an integral type.

QAC Messages: 0303, 0305, 0306

**MISRA-C:2004 Rule 11.4** (Advisory)

A cast should not be performed between a pointer to object type and a different pointer to object type.

QAC Messages: 0310

**MISRA-C:2004 Rule 11.5** (Required)

A cast shall not be performed that removes any const or volatile qualification from the type addressed by a pointer.

QAC Messages: 0311, 0312

**MISRA-C:2004 Rule 12.1** (Advisory)

Limited dependence should be placed on C's operator precedence rules in expressions.

QAC Messages: 3389, 3391, 3392, 3393, 3394, 3395, 3396, 3397

**MISRA-C:2004 Rule 12.2** (Required)

The value of an expression shall be the same under any order of evaluation that the standard permits.

QAC Messages: 0400, 0401, 0402, 0403

**MISRA-C:2004 Rule 12.3** (Required)

The sizeof operator shall not be used on expressions that contain side effects.

QAC Messages: 3307

**MISRA-C:2004 Rule 12.4** (Required)

The right hand operand of a logical && or || operator shall not contain side effects.

QAC Messages: 3415

**MISRA-C:2004 Rule 12.5** (Required)

The operands of a logical && or || shall be primary-expressions.

QAC Messages: 3398, 3399, 3400

**MISRA-C:2004 Rule 12.6** (Advisory)

The operands of logical operators (&&, || and !) should be effectively Boolean. Expressions that are effectively Boolean should not be used as operands to operators other than (&&, || and !).

QAC Messages: 4101, 4102, 4103, 4104, 4105, 4106, 4107, 4108, 4109, 4110, 4111, 4112, 4113, 4114

**MISRA-C:2004 Rule 12.7** (Required)

Bitwise operators shall not be applied to operands whose underlying type is signed.

QAC Messages: 0502, 4130, 4131

**MISRA-C:2004 Rule 12.8** (Required)

The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand.

QAC Messages: 0499, 0500, 0501

**MISRA-C:2004 Rule 12.9** (Required)

The unary minus operator shall not be applied to an expression whose underlying type is unsigned.

QAC Messages: 3101, 3102

**MISRA-C:2004 Rule 12.10** (Required)

The comma operator shall not be used.

QAC Messages: 3417, 3418

**MISRA-C:2004 Rule 12.11** (Advisory)

Evaluation of constant unsigned integer expressions should not lead to wrap-around.

QAC Messages: 3302, 3303, 3304

**MISRA-C:2004 Rule 12.12** (Required)

The underlying bit representations of floating-point values shall not be used.

QAC Messages: 3629

**MISRA-C:2004 Rule 12.13** (Advisory)

The increment (++) and decrement (--) operators should not be mixed with other operators in an expression.

QAC Messages: 3440

**MISRA-C:2004 Rule 13.1** (Required)

Assignment operators shall not be used in expressions that yield a Boolean value.

QAC Messages: 3326

**MISRA-C:2004 Rule 13.2** (Advisory)

Tests of a value against zero should be made explicit, unless the operand is effectively Boolean.

QAC Messages: 3344

**MISRA-C:2004 Rule 13.3** (Required)

Floating-point expressions shall not be tested for equality or inequality.

QAC Messages: 3341

**MISRA-C:2004 Rule 13.4** (Required)

The controlling expression of a for statement shall not contain any objects of floating type.

QAC Messages: 3340

**MISRA-C:2004 Rule 13.5** (Required)

The three expressions of a for statement shall be concerned only with loop control.

QAC Messages: 2462, 2463

**MISRA-C:2004 Rule 13.6** (Required)

Numeric variables being used within a for loop for iteration counting shall not be modified in the body of the loop.

QAC Messages: 2469

**MISRA-C:2004 Rule 13.7** (Required)

Boolean operations whose results are invariant shall not be permitted.

QAC Messages: 3355, 3356, 3357, 3358, 3359, 3360

**MISRA-C:2004 Rule 14.1** (Required)

There shall be no unreachable code.

QAC Messages: 0594, 0689, 1460, 1503, 2008, 2018, 3201, 3219, 3325

**MISRA-C:2004 Rule 14.2** (Required)

All non-null statements shall either (i) have at least one side-effect however executed, or (ii) cause control flow to change.

QAC Messages: 3110, 3112, 3404, 3425, 3426, 3427

**MISRA-C:2004 Rule 14.3** (Required)

Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment provided that the first character following the null statement is a white-space character.  
QAC Messages: 3109

**MISRA-C:2004 Rule 14.4 (Required)**

The goto statement shall not be used.

QAC Messages: 2001

**MISRA-C:2004 Rule 14.5 (Required)**

The continue statement shall not be used.

QAC Messages: 0770

**MISRA-C:2004 Rule 14.6 (Required)**

For any iteration statement there shall be at most one break statement used for loop termination.

QAC Messages: 0771, 3333

**MISRA-C:2004 Rule 14.7 (Required)**

A function shall have a single point of exit at the end of the function.

QAC Messages: 2006

**MISRA-C:2004 Rule 14.8 (Required)**

The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement.

QAC Messages: 2212, 2214

**MISRA-C:2004 Rule 14.9 (Required)**

An if (expression) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement.

QAC Messages: 2212, 2214, 3402

**MISRA-C:2004 Rule 14.10 (Required)**

All if ... else if constructs shall be terminated with an else clause.

QAC Messages: 2004

**MISRA-C:2004 Rule 15.1 (Required)**

A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement.

QAC Messages: 2019

**MISRA-C:2004 Rule 15.2 (Required)**

An unconditional break statement shall terminate every non-empty switch clause.

QAC Messages: 2003

**MISRA-C:2004 Rule 15.3 (Required)**

The final clause of a switch statement shall be the default clause.

QAC Messages: 2002, 2009

**MISRA-C:2004 Rule 15.4 (Required)**

A switch expression shall not represent a value that is effectively Boolean.

QAC Messages: 0735

**MISRA-C:2004 Rule 15.5 (Required)**

Every switch statement shall have at least one case clause.

QAC Messages: 3315



**MISRA-C:2004 Rule 16.1 (Required)**

Functions shall not be defined with a variable number of arguments.

QAC Messages: 5069

**MISRA-C:2004 Rule 16.2 (Required)**

Functions shall not call themselves, either directly or indirectly.

QAC Messages: 1520, 3670

**MISRA-C:2004 Rule 16.3 (Required)**

Identifiers shall be given for all of the parameters in a function prototype declaration.

QAC Messages: 0652

**MISRA-C:2004 Rule 16.4 (Required)**

The identifiers used in the declaration and definition of a function shall be identical.

QAC Messages: 1330, 1334

**MISRA-C:2004 Rule 16.5 (Required)**

Functions with no parameters shall be declared with parameter type void.

QAC Messages: 3001, 3007

**MISRA-C:2004 Rule 16.6 (Required)**

The number of arguments passed to a function shall match the number of parameters.

QAC Messages: 0422, 0423, 3319

**MISRA-C:2004 Rule 16.7 (Advisory)**

A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object.

QAC Messages: 3673

**MISRA-C:2004 Rule 16.8 (Required)**

All exit paths from a function with non-void return type shall have an explicit return statement with an expression.

QAC Messages: 0744, 0745, 3113, 3114

**MISRA-C:2004 Rule 16.9 (Required)**

A function identifier shall only be used with either a preceding &, or with a parenthesised parameter list, which may be empty.

QAC Messages: 0428

**MISRA-C:2004 Rule 16.10 (Required)**

If a function returns error information, then that error information shall be tested.

QAC Messages: 3200

**MISRA-C:2004 Rule 17.1 (Required)**

Pointer arithmetic shall only be applied to pointers that address an array or array element.

**MISRA-C:2004 Rule 17.2 (Required)**

Pointer subtraction shall only be applied to pointers that address elements of the same array.

**MISRA-C:2004 Rule 17.3 (Required)**

>, >=, <, <= shall not be applied to pointer types except where they point to the same array.

**MISRA-C:2004 Rule 17.4 (Required)**

Array indexing shall be the only allowed form of pointer arithmetic.

QAC Messages: 0488, 0489

**MISRA-C:2004 Rule 17.5** (Advisory)

The declaration of objects should contain no more than 2 levels of pointer indirection.

QAC Messages: 5102

**MISRA-C:2004 Rule 17.6** (Required)

The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist.

QAC Messages: 3217, 3225, 3230, 4140

**MISRA-C:2004 Rule 18.1** (Required)

All structure and union types shall be complete at the end of a translation unit.

QAC Messages: 0544, 0545, 0623, 0636, 3313

**MISRA-C:2004 Rule 18.2** (Required)

An object shall not be assigned to an overlapping object.

**Enforcement:** NSE

**MISRA-C:2004 Rule 18.3** (Required)

An area of memory shall not be reused for unrelated purposes.

**Enforcement:** NSE

**MISRA-C:2004 Rule 18.4** (Required)

Unions shall not be used.

QAC Messages: 0750, 0759

**MISRA-C:2004 Rule 19.1** (Advisory)

#include statements in a file should only be preceded by other preprocessor directives or comments.

QAC Messages: 5087

**MISRA-C:2004 Rule 19.2** (Advisory)

Non-standard characters should not occur in header file names in #include directives.

QAC Messages: 0813, 0814, 0831

**MISRA-C:2004 Rule 19.3** (Required)

The #include directive shall be followed by either a <filename> or "filename" sequence.

QAC Messages: 0809

**MISRA-C:2004 Rule 19.4** (Required)

C macros shall only expand to a braced initialiser, a constant, a parenthesised expression, a type qualifier, a storage class specifier, or a do-while-zero construct.

QAC Messages: 3411, 3412, 3413, 3431, 3458, 3460, 3461

**MISRA-C:2004 Rule 19.5** (Required)

Macros shall not be #define'd or #undef'd within a block.

QAC Messages: 0842

**MISRA-C:2004 Rule 19.6** (Required)

#undef shall not be used.

QAC Messages: 0841

**MISRA-C:2004 Rule 19.7** (Advisory)

A function should be used in preference to a function-like macro.

QAC Messages: 3453

**MISRA-C:2004 Rule 19.8** (Required)

A function-like macro shall not be invoked without all of its arguments.

QAC Messages: 0850, 0856

**MISRA-C:2004 Rule 19.9** (Required)

Arguments to a function-like macro shall not contain tokens that look like preprocessing directives.

QAC Messages: 0853

**MISRA-C:2004 Rule 19.10** (Required)

In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##.

QAC Messages: 3410

**MISRA-C:2004 Rule 19.11** (Required)

All macro identifiers in preprocessor directives shall be defined before use, except in #ifdef and #ifndef preprocessor directives and the defined() operator.

QAC Messages: 3332

**MISRA-C:2004 Rule 19.12** (Required)

There shall be at most one occurrence of the # or ## preprocessor operators in a single macro definition.

QAC Messages: 0880, 0881

**MISRA-C:2004 Rule 19.13** (Advisory)

The # and ## preprocessor operators should not be used.

QAC Messages: 0341, 0342

**MISRA-C:2004 Rule 19.14** (Required)

The defined preprocessor operator shall only be used in one of the two standard forms.

QAC Messages: 0885, 0887, 0888

**MISRA-C:2004 Rule 19.15** (Required)

Precautions shall be taken in order to prevent the contents of a header file being included twice.

QAC Messages: 0883

**MISRA-C:2004 Rule 19.16** (Required)

Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor.

QAC Messages: 3115

**MISRA-C:2004 Rule 19.17** (Required)

All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if or #ifdef directive to which they are related.

QAC Messages: 3317, 3318

**MISRA-C:2004 Rule 20.1** (Required)

Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined.

QAC Messages: 0836, 0848, 0854, 3439, 5114, 5214

**MISRA-C:2004 Rule 20.2** (Required)

The names of standard library macros, objects and functions shall not be reused.

QAC Messages: 0602, 5115

**MISRA-C:2004 Rule 20.3** (Required)

The validity of values passed to library functions shall be checked.

**MISRA-C:2004 Rule 20.4** (Required)

Dynamic heap memory allocation shall not be used.

QAC Messages: 5118

**MISRA-C:2004 Rule 20.5** (Required)

The error indicator `errno` shall not be used.

QAC Messages: 5119

**MISRA-C:2004 Rule 20.6** (Required)

The macro `offsetof`, in library `<stddef.h>`, shall not be used.

QAC Messages: 5120

**MISRA-C:2004 Rule 20.7** (Required)

The `setjmp` macro and the `longjmp` function shall not be used.

QAC Messages: 5122

**MISRA-C:2004 Rule 20.8** (Required)

The signal handling facilities of `<signal.h>` shall not be used.

QAC Messages: 5123

**MISRA-C:2004 Rule 20.9** (Required)

The input/output library `<stdio.h>` shall not be used in production code.

QAC Messages: 5124

**MISRA-C:2004 Rule 20.10** (Required)

The library functions `atof`, `atoi` and `atol` from library `<stdlib.h>` shall not be used.

QAC Messages: 5125

**MISRA-C:2004 Rule 20.11** (Required)

The library functions `abort`, `exit`, `getenv` and `system` from library `<stdlib.h>` shall not be used.

QAC Messages: 5126

**MISRA-C:2004 Rule 20.12** (Required)

The time handling functions of library `<time.h>` shall not be used.

QAC Messages: 5127

**MISRA-C:2004 Rule 21.1** (Required)

Minimisation of run-time failures shall be ensured by the use of at least one of (a) static analysis tools/techniques; (b) dynamic analysis tools/techniques; (c) explicit coding of checks to handle run-time faults.

QAC Messages: 0272, 0273, 0290, 0291, 0296, 0297, 0585, 0587, 3372, 3382, 3685, 3689