
# MISRA C:2012

## Addendum 1 — Rule Mappings

March 2013


First published March 2013 by MIRA Limited
Watling Street
Nuneaton
Warwickshire
CV10 0TU
UK

www.misra.org.uk



ISBN 978-1-906400-12-5 PDF

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

# MISRA C:2012 Addendum: Rule Mappings

## MISRA C:2004 to MISRA C:2012 rule mapping

| MISRA C:2004 | MISRA C:2012 | Significant changes from MISRA C:2004 for C90 code |
|---|---|---|
| Rule 1.1 (required) | Rule 1.1 (required) | The requirement that code conform to a version of the standard has moved into an introductory section. MISRA C:2004 Rule 1.1 had an implied ban on extensions. MISRA C:2012 Rule 1.1 permits extensions, but they are restricted by the new advisory Rule 1.2. |
| | Rule 1.2 (advisory) | |
| Rule 1.2 (required) | Rule 1.3 (required) | Relaxed to permit unspecified behaviour that is not considered critical. MISRA C:2012 Appendix H lists the issues that are covered. |
| Rule 1.3 (required) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 1.4 (required) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 1.5 (advisory) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 2.1 (required) | Dir 4.2 (advisory) | Tightened to require documentation of assembly usage. |
| | Dir 4.3 (required) | |
| Rule 2.2 (required) | Rule 1.2 (advisory) | Relaxed to permit // comments in C99 code. Downgraded to advisory for C90. |
| Rule 2.3 (required) | Rule 3.1 (required) | |
| Rule 2.4 (advisory) | Dir 4.4 (advisory) | Changed to a directive as MISRA C:2012 does not precisely describe what constitutes code in a comment. |
| Rule 3.1 (required) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 3.2 (required) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 3.3 (advisory) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. It is not an issue for C99 because the standard specifies the behaviour. |
| Rule 3.4 (required) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 3.5 (required) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 3.6 (required) | Deleted | This issue is now covered by the Introduction to the Rules section. |
| Rule 4.1 (required) | Rule 4.1 (required) | Relaxed to permit octal and hexadecimal escapes providing that they are properly terminated. C90 Undefined 11 (an undefined escape sequence is used) is now covered by Rule 1.3. |
| Rule 4.2 (required) | Rule 4.2 (advisory) | Downgraded to advisory because there is no undefined or unspecified behaviour associated with trigraphs. There are certain source character sets in which it would be impossible to write C programs without trigraphs. |
| Rule 5.1 (required) | Rule 5.1 (required) | Relaxed to permit the identifiers to be distinct up to the limit allowed by the implementation. |
| | Rule 5.2 (required) | |
| | Rule 5.3 (required) | |
| | Rule 5.4 (required) | |
| | Rule 5.5 (required) | |
| Rule 5.2 (required) | Rule 5.3 (required) | |

| MISRA C:2004 | MISRA C:2012 | Significant changes from MISRA C:2004 for C90 code |
|---|---|---|
| Rule 5.3 (required) | Rule 5.6 (required) | Relaxed to permit tag names that are the same as *typedef* names. |
| Rule 5.4 (required) | Rule 5.7 (required) | Relaxed to permit tag names that are the same as *typedef* names. |
| Rule 5.5 (advisory) | Rule 5.8 (required) | Relaxed to permit *static* identifiers declared in a block to have the same name as other identifiers. |
| | Rule 5.9 (advisory) | |
| Rule 5.6 (advisory) | Deleted | Relaxed to permit identifiers such as labels and enumeration constants that are not in the same scope to have the same identifier. |
| Rule 5.7 (advisory) | Deleted | Relaxed to permit identifiers such as structure members names in different structures. |
| Rule 6.1 (required) | Rule 10.1 (required) | Relaxed to allow addition of plain *char* and a different integer type, and subtraction of two plain *char*. |
| | Rule 10.2 (required) | |
| | Rule 10.3 (required) | |
| | Rule 10.4 (required) | |
| Rule 6.2 (required) | Rule 10.1 (required) | |
| | Rule 10.3 (required) | |
| | Rule 10.4 (required) | |
| Rule 6.3 (advisory) | Dir 4.6 (advisory) | |
| Rule 6.4 (required) | Rule 6.1 (required) | Relaxed to permit other legal C99 bit-field types |
| Rule 6.5 (required) | Rule 6.2 (required) | Relaxed to permit unnamed bit-fields with a signed type to be single-bit. |
| Rule 7.1 (required) | Rule 4.1 (required) | Relaxed to permit octal escape sequences providing they are properly terminated. |
| | Rule 7.1 (required) | |
| Rule 8.1 (required) | Rule 8.2 (required) | Tightened to require all function types to be in prototype form with named parameters.<br>Relaxed to permit function definitions with no separate declaration if they have internal linkage. |
| | Rule 8.4 (required) | |
| | Rule 17.3 (mandatory) | |
| Rule 8.2 (required) | Rule 8.1 (required) | Tightened to require all types to be explicitly stated, e.g. structure members. |
| Rule 8.3 (required) | Rule 8.3 (required) | |
| Rule 8.4 (required) | Rule 8.3 (required) | Tightened to require identical object types |
| Rule 8.5 (required) | Deleted | This rule has been deleted because some programming paradigms require the presence of executable code in a header file. It is also desirable for inline functions to appear in header files so as to avoid undefined issues. The problems associated with multiple declaration/definition of objects/functions are dealt with by other rules. |
| Rule 8.6 (required) | Deleted | Relaxed to permit declarations using the *extern* keyword at block scope, but more specific problems are caught by other rules such as:<br>• Rule 1.3: Undefined behaviour due to use of the *static* keyword<br>• Rule 8.3: Compatibility of multiply-declared/defined objects/functions |

| MISRA C:2004 | MISRA C:2012 | Significant changes from MISRA C:2004 for C90 code |
|---|---|---|
| Rule 8.7 (required) | Rule 8.9 (advisory) | Downgraded to advisory because there are instances where following this rule makes it impossible to perform adequate unit testing. |
| Rule 8.8 (required) | Rule 8.5 (required) | |
| Rule 8.9 (required) | Rule 8.6 (required) | |
| Rule 8.10 (required) | Rule 8.7 (advisory) | Downgraded to advisory because there are instances where following this rule makes it impossible to perform adequate unit testing. |
| Rule 8.11 (required) | Rule 8.8 (required) | |
| Rule 8.12 (required) | Rule 8.11 (advisory) | Tightened to require inclusion of the size for arrays with external linkage, even if defined implicitly by initialization. |
| Rule 9.1 (required) | Rule 9.1 (mandatory) | |
| Rule 9.2 (required) | Rule 9.2 (required) | Relaxed to permit use of `{0}` in subarrays or sub-structures.<br>Relaxed to permit initialization with a string literal. |
| | Rule 9.3 (required) | |
| Rule 9.3 (required) | Rule 8.12 (required) | Relaxed to permit partial initialization providing all members are unique |
| Rule 10.1 (required) | Rule 10.3 (required) | Tightened to include plain *char*, Boolean and enumerated types.<br>Relaxed to permit implicit widening conversions on function arguments or return values.<br>Relaxed to permit signed integer constants to be used in unsigned contexts, providing the value is within the unsigned range. |
| | Rule 10.4 (required) | |
| | Rule 10.6 (required) | |
| | Rule 10.7 (required) | |
| Rule 10.2 (required) | Rule 10.3 (required) | Relaxed to permit implicit widening conversions on function arguments or return values. |
| | Rule 10.4 (required) | |
| | Rule 10.6 (required) | |
| | Rule 10.7 (required) | |
| Rule 10.3 (required) | Rule 10.8 (required) | Tightened to include plain *char*, Boolean and enumerated types. |
| Rule 10.4 (required) | Rule 10.8 (required) | |
| Rule 10.5 (required) | Deleted | This rule has been deleted as it did not adequately address the underlying problem. The enhanced type rules address some of the issues previously covered by this rule. |
| Rule 10.6 (required) | Rule 7.2 (required) | |
| Rule 11.1 (required) | Rule 11.1 (required) | Tightened to include conversions to/from integral types.<br>Relaxed to permit conversions from a *null pointer constant*, and conversion to *void*. |
| Rule 11.2 (required) | Rule 11.1 (required) | Tightened to include conversions from *void **, and to/from Boolean, plain *char* and enumeration types. |
| | Rule 11.2 (required) | |
| | Rule 11.5 (advisory) | |
| | Rule 11.7 (required) | |

| MISRA C:2004 | MISRA C:2012 | Significant changes from MISRA C:2004 for C90 code |
|---|---|---|
| Rule 11.3 (advisory) | Rule 11.1 (required) | |
| | Rule 11.2 (required) | |
| | Rule 11.4 (advisory) | |
| | Rule 11.6 (required) | |
| Rule 11.4 (advisory) | Rule 11.3 (required) | Upgraded to required. Relaxed to permit casts that convert into pointer to character types. |
| Rule 11.5 (required) | Rule 11.8 (required) | |
| Rule 12.1 (advisory) | Rule 12.1 (advisory) | Tightened MISRA C:2004 Rule 12.1 by describing the rule's advice in terms of the language syntax. |
| Rule 12.2 (required) | Rule 13.2 (required) | |
| Rule 12.3 (required) | Rule 13.6 (mandatory) | Relaxed to permit *sizeof* to be applied to an expression whose only *side effect* is to read from a *volatile* object. |
| Rule 12.4 (required) | Rule 13.5 (required) | |
| Rule 12.5 (required) | Rule 12.1 (advisory) | Relaxed to permit postfix, unary and cast operands of `&&` and `\|\|` without brackets. |
| Rule 12.6 (advisory) | Rule 10.1 (required) | |
| Rule 12.7 (required) | Rule 10.1 (required) | |
| Rule 12.8 (required) | Rule 12.2 (required) | |
| Rule 12.9 (required) | Rule 10.1 (required) | |
| Rule 12.10 (required) | Rule 12.3 (advisory) | Downgraded to advisory, as there are occasions where there are no easy ways in which to achieve the same effect. |
| Rule 12.11 (advisory) | Rule 12.4 (advisory) | |
| Rule 12.12 (required) | Dir 1.1 (required) | Changed to a directive as it is not a statically checkable rule. |
| Rule 12.13 (advisory) | Rule 13.3 (advisory) | Relaxed to permit ++ and −− to be mixed with other operators, provided that the ++ or −− is the only source of side-effects. |
| Rule 13.1 (required) | Rule 13.4 (advisory) | Tightened so that the result of an assignment operator can not be used anywhere. For example, `a[x=y]=x;`<br>Downgraded to advisory as the more important issues are addressed by other MISRA C:2012 required rules. For example:<br>• Rule 13.2 covers evaluation order of *side effects*. |
| Rule 13.2 (advisory) | Rule 14.4 (required) | |
| Rule 13.3 (required) | Dir 1.1 (required) | Focussing on one aspect of floating-point arithmetic (as in MISRA C:2004 Rule 13.3) attached too much importance to that aspect and might give users a false sense of confidence. |
| Rule 13.4 (required) | Rule 14.1 (required) | Relaxed to permit objects with floating-point type in the controlling expression, providing they are not a *loop counter*. |
| Rule 13.5 (required) | Rule 14.2 (required) | |
| Rule 13.6 (required) | Rule 14.2 (required) | |
| Rule 13.7 (required) | Rule 14.3 (required) | Relaxed to permit invariant Boolean expressions in other contexts, such as assignments. This was the original intention for MISRA C:2004 Rule 13.7. |
| Rule 14.1 (required) | Rule 2.1 (required) | |

| MISRA C:2004 | MISRA C:2012 | Significant changes from MISRA C:2004 for C90 code |
|---|---|---|
| Rule 14.2 (required) | Rule 2.2 (required) | Tightened to include assignments to variables that are not subsequently read. |
| Rule 14.3 (required) | Rule 15.6 (required) | Relaxed to permit null statements in more places.<br>MISRA C:2012 Rule 15.6 focuses on the detection of problems such as `if(C);{...}` . |
| Rule 14.4 (required) | Rule 15.1 (advisory) | Downgraded to advisory but there are additional required rules in case Rule 14.1 is not applied. Historically, there has been much debate over the advisability of using constructs such as *goto*. The intervening years have also seen an improvement in tools, which permit a better analysis of code with more complicated structures. |
| | Rule 15.2 (required) | |
| | Rule 15.3 (required) | |
| Rule 14.5 (required) | Deleted | Historically, there has been much debate over the advisability of using constructs such as *goto* and *continue*. The rationale given in MISRA C:2004 Rule 14.5 was weak — it cites structured programming, but *continue* is a structured programming construct. The intervening years have also seen an improvement in tools, which permit a better analysis of code with more complicated structures. |
| Rule 14.6 (required) | Rule 15.4 (advisory) | Relaxed to include using *goto* for the same purpose.<br>*Note:* Use of *goto* assumes that rule MISRA C:2012 Rule 14.1 has not been applied.<br>Downgraded to advisory. |
| Rule 14.7 (required) | Rule 15.5 (advisory) | Downgraded to advisory as there may be good reasons for having multiple exit paths (as for multiple *break* statements). |
| Rule 14.8 (required) | Rule 15.6 (required) | |
| Rule 14.9 (required) | Rule 15.6 (required) | |
| Rule 14.10 (required) | Rule 15.7 (required) | |
| Rule 15.0 (required) | Rule 16.1 (required) | Relaxed to permit the *default* label to occur as either first or last *switch-clause*. |
| Rule 15.1 (required) | Rule 16.2 (required) | |
| Rule 15.2 (required) | Rule 16.3 (required) | |
| Rule 15.3 (required) | Rule 16.4 (required) | Relaxed to permit the *default* label to occur as either first or last *switch-clause*. |
| | Rule 16.5 (required) | |
| Rule 15.4 (required) | Rule 16.7 (required) | |
| Rule 15.5 (required) | Rule 16.6 (required) | |
| Rule 16.1 (required) | Rule 17.1 (required) | Tightened to include all uses of the features provided by `<stdarg.h>`. |
| Rule 16.2 (required) | Rule 17.2 (required) | |
| Rule 16.3 (required) | Rule 8.2 (required) | |
| Rule 16.4 (required) | Rule 8.3 (required) | |
| Rule 16.5 (required) | Rule 8.2 (required) | |
| Rule 16.6 (required) | Rule 8.2 (required) | |
| | Rule 17.3 (mandatory) | |
| Rule 16.7 (advisory) | Rule 8.13 (advisory) | |
| Rule 16.8 (required) | Rule 17.4 (mandatory) | |

| MISRA C:2004 | MISRA C:2012 | Significant changes from MISRA C:2004 for C90 code |
|---|---|---|
| Rule 16.9 (required) | Deleted | Relaxed to permit the use of function name on its own. The issues raised by MISRA C:2004 Rule 16.9 are now covered by the set of type checking rules: MISRA C:2012 Rules 10.1–10.4. |
| Rule 16.10 (required) | Dir 4.7 (required) | Changed to directive as MISRA C:2012 does not specify which functions return error information or how the check should occur. |
| Rule 17.1 (required) | Rule 18.1 (required) | |
| Rule 17.2 (required) | Rule 18.2 (required) | |
| Rule 17.3 (required) | Rule 18.3 (required) | |
| Rule 17.4 (required) | Rule 18.4 (advisory) | Relaxed to permit ++ and –– on pointers.<br>Downgraded to advisory. |
| Rule 17.5 (advisory) | Rule 18.5 (advisory) | |
| Rule 17.6 (required) | Rule 18.6 (required) | |
| Rule 18.1 (required) | Rule 1.3 (required) | Relaxed to permit incomplete types, providing undefined behaviour does not occur. |
| Rule 18.2 (required) | Rule 19.1 (mandatory) | |
| Rule 18.3 (required) | Deleted | This rule has been deleted because its interpretation is subjective and it is not statically checkable. |
| Rule 18.4 (required) | Rule 19.2 (advisory) | Downgraded to advisory. |
| Rule 19.1 (advisory) | Rule 20.1 (advisory) | |
| Rule 19.2 (advisory) | Rule 20.2 (required) | Tightened to include use of \.<br>Upgraded to required. |
| Rule 19.3 (required) | Rule 20.3 (required) | |
| Rule 19.4 (required) | Rule 20.4 (required) | Relaxed so that the only thing not permitted is to define a macro with the same name as a keyword.<br>Other MISRA C:2012 rules covers the issues that MISRA C:2004 Rule 19.4 was addressing. For example:<br>• Rule 12.1 requires use of parenthesis to make the precedence of an expression clear.<br>• Rule 14.6 requires that a macro with more than 1 statement does not get expanded in an iteration or selection statements.<br>• Rules 10.1–10.4 prevent<br>`#define INTPTR int *`<br>`INTPTR a, b;`<br>The user could not use `b` as if it were a pointer without violating a MISRA type check rule. |
| Rule 19.5 (required) | Deleted | This rule dealt with a stylistic issue. |
| Rule 19.6 (required) | Rule 20.5 (advisory) | Downgraded to advisory. |
| Rule 19.7 (advisory) | Dir 4.9 (advisory) | |
| Rule 19.8 (required) | Rule 1.3 (required) | In C90 the behaviour is undefined and is caught by rule MISRA C:2012 Rule 1.3. |
| Rule 19.9 (required) | Rule 20.6 (required) | |

| MISRA C:2004 | MISRA C:2012 | Significant changes from MISRA C:2004 for C90 code |
|---|---|---|
| Rule 19.10 (required) | Rule 20.7 (required) | Relaxed so that parenthesis are only required around parameters that are used in an expression context. The strict application MISRA C:2004 Rule 19.10 could produce syntax errors. For example, in:<br>`#define GET_MEMBER(S,M) (S).M`<br>it is a constraint error to enclose `M` in parentheses. |
| Rule 19.11 (required) | Rule 20.9 (required) | |
| Rule 19.12 (required) | Rule 20.11 (required) | Relaxed to cover the undefined behaviour without being unnecessarily restrictive. |
| Rule 19.13 (advisory) | Rule 20.10 (advisory) | |
| Rule 19.14 (required) | Rule 1.3 (required) | The undefined behaviour is covered by MISRA C:2012 Rule 1.3. |
| Rule 19.15 (required) | Dir 4.10 (required) | Changed to a directive as MISRA C:2012 does not specify the form that the precautions should take. |
| Rule 19.16 (required) | Rule 20.13 (required) | |
| Rule 19.17 (required) | Rule 20.14 (required) | |
| Rule 20.1 (required) | Rule 21.1 (required) | Relaxed to permit *#define* on Future Identifiers . |
| Rule 20.2 (required) | Rule 21.2 (required) | |
| Rule 20.3 (required) | Dir 4.11 (required) | |
| Rule 20.4 (required) | Rule 21.3 (required) | |
| | Dir 4.12 (required) | |
| Rule 20.5 (required) | Dir 1.1 (required) | Relaxed to permit the use of *errno*, providing evidence of its use in the program is documented. |
| Rule 20.6 (required) | Rule 1.3 (required) | Relaxed to permit the use of *offsetof* except where undefined behaviour might occur. |
| Rule 20.7 (required) | Rule 21.4 (required) | |
| Rule 20.8 (required) | Rule 21.5 (required) | |
| Rule 20.9 (required) | Rule 21.6 (required) | Tightened to include the I/O routines in `<wchar.h>`. |
| Rule 20.10 (required) | Rule 21.7 (required) | Tightened to include *atoll*. |
| Rule 20.11 (required) | Rule 21.8 (required) | |
| Rule 20.12 (required) | Rule 21.10 (required) | Tightened to include *wcsftime* in `<wchar.h>`. |
| Rule 21.1 (required) | Dir 4.1 (required) | Changed to a directive as it is not a statically checkable rule. |

# MISRA C:2012 to MISRA C:2004 rule mapping

| MISRA C:2012 | MISRA C:2004 |
|---|---|
| Dir 1.1 (required) | Rule 1.3 (required) |
| | Rule 1.4 (required) |
| | Rule 1.5 (advisory) |
| | Rule 3.1 (required) |
| | Rule 3.2 (required) |
| | Rule 3.3 (advisory) |
| | Rule 3.4 (required) |
| | Rule 3.5 (required) |
| | Rule 12.12 (required) |
| | Rule 13.3 (required) |
| | Rule 20.5 (required) |
| Dir 2.1 (required) | New |
| Dir 3.1 (required) | New |
| Dir 4.1 (required) | Rule 21.1 (required) |
| Dir 4.2 (advisory) | Rule 2.1 (required) |
| Dir 4.3 (required) | Rule 2.1 (required) |
| Dir 4.4 (advisory) | Rule 2.4 (advisory) |
| Dir 4.5 (advisory) | New |
| Dir 4.6 (advisory) | Rule 6.3 (advisory) |
| Dir 4.7 (required) | Rule 16.10 (required) |
| Dir 4.8 (advisory) | New |
| Dir 4.9 (advisory) | Rule 19.7 (advisory) |
| Dir 4.10 (required) | Rule 19.15 (required) |
| Dir 4.11 (required) | Rule 20.3 (required) |
| Dir 4.12 (required) | Rule 20.4 (required) |
| Dir 4.13 (advisory) | New |
| Rule 1.1 (required) | Rule 1.1 (required) |
| Rule 1.2 (advisory) | Rule 1.1 (required) |
| | Rule 2.2 (required) |
| Rule 1.3 (required) | Rule 1.2 (required) |
| | Rule 18.1 (required) |
| | Rule 19.8 (required) |
| | Rule 19.14 (required) |
| | Rule 20.6 (required) |
| Rule 2.1 (required) | Rule 14.1 (required) |
| Rule 2.2 (required) | Rule 14.2 (required) |
| Rule 2.3 (advisory) | New |

| MISRA C:2012 | MISRA C:2004 |
|---|---|
| Rule 2.4 (advisory) | New |
| Rule 2.5 (advisory) | New |
| Rule 2.6 (advisory) | New |
| Rule 2.7 (advisory) | New |
| Rule 3.1 (required) | Rule 2.3 (required) |
| Rule 3.2 (required) | New |
| Rule 4.1 (required) | Rule 4.1 (required) |
| | Rule 7.1 (required) |
| Rule 4.2 (advisory) | Rule 4.2 (required) |
| Rule 5.1 (required) | Rule 5.1 (required) |
| Rule 5.2 (required) | Rule 5.1 (required) |
| Rule 5.3 (required) | Rule 5.1 (required) |
| | Rule 5.2 (required) |
| Rule 5.4 (required) | Rule 5.1 (required) |
| Rule 5.5 (required) | Rule 5.1 (required) |
| Rule 5.6 (required) | Rule 5.3 (required) |
| Rule 5.7 (required) | Rule 5.4 (required) |
| Rule 5.8 (required) | Rule 5.5 (advisory) |
| Rule 5.9 (advisory) | Rule 5.5 (advisory) |
| Rule 6.1 (required) | Rule 6.4 (required) |
| Rule 6.2 (required) | Rule 6.5 (required) |
| Rule 7.1 (required) | Rule 7.1 (required) |
| Rule 7.2 (required) | Rule 10.6 (required) |
| Rule 7.3 (required) | New |
| Rule 7.4 (required) | New |
| Rule 8.1 (required) | Rule 8.2 (required) |
| Rule 8.2 (required) | Rule 8.1 (required) |
| | Rule 16.3 (required) |
| | Rule 16.5 (required) |
| | Rule 16.6 (required) |
| Rule 8.3 (required) | Rule 8.3 (required) |
| | Rule 8.4 (required) |
| | Rule 16.4 (required) |
| Rule 8.4 (required) | Rule 8.1 (required) |
| Rule 8.5 (required) | Rule 8.8 (required) |
| Rule 8.6 (required) | Rule 8.9 (required) |
| Rule 8.7 (advisory) | Rule 8.10 (required) |
| Rule 8.8 (required) | Rule 8.11 (required) |

| MISRA C:2012 | MISRA C:2004 |
|---|---|
| Rule 8.9 (advisory) | Rule 8.7 (required) |
| Rule 8.10 (required) | New |
| Rule 8.11 (advisory) | Rule 8.12 (required) |
| Rule 8.12 (required) | Rule 9.3 (required) |
| Rule 8.13 (advisory) | Rule 16.7 (advisory) |
| Rule 8.14 (required) | New |
| Rule 9.1 (mandatory) | Rule 9.1 (required) |
| Rule 9.2 (required) | Rule 9.2 (required) |
| Rule 9.3 (required) | Rule 9.2 (required) |
| Rule 9.4 (required) | New |
| Rule 9.5 (required) | New |
| Rule 10.1 (required) | Rule 6.1 (required) |
| | Rule 6.2 (required) |
| | Rule 12.6 (advisory) |
| | Rule 12.7 (required) |
| | Rule 12.9 (required) |
| Rule 10.2 (required) | Rule 6.1 (required) |
| Rule 10.3 (required) | Rule 6.1 (required) |
| | Rule 6.2 (required) |
| | Rule 10.1 (required) |
| | Rule 10.2 (required) |
| Rule 10.4 (required) | Rule 6.1 (required) |
| | Rule 6.2 (required) |
| | Rule 10.1 (required) |
| | Rule 10.2 (required) |
| Rule 10.5 (advisory) | New |
| Rule 10.6 (required) | Rule 10.1 (required) |
| | Rule 10.2 (required) |
| Rule 10.7 (required) | Rule 10.1 (required) |
| | Rule 10.2 (required) |
| Rule 10.8 (required) | Rule 10.3 (required) |
| | Rule 10.4 (required) |
| Rule 11.1 (required) | Rule 11.1 (required) |
| | Rule 11.2 (required) |
| | Rule 11.3 (advisory) |
| Rule 11.2 (required) | Rule 11.2 (required) |
| | Rule 11.3 (advisory) |
| Rule 11.3 (required) | Rule 11.4 (advisory) |

| MISRA C:2012 | MISRA C:2004 |
|---|---|
| Rule 11.4 (advisory) | Rule 11.3 (advisory) |
| Rule 11.5 (advisory) | Rule 11.2 (required) |
| Rule 11.6 (required) | Rule 11.3 (advisory) |
| Rule 11.7 (required) | Rule 11.2 (required) |
| Rule 11.8 (required) | Rule 11.5 (required) |
| Rule 11.9 (required) | New |
| Rule 12.1 (advisory) | Rule 12.1 (advisory) |
| | Rule 12.5 (required) |
| Rule 12.2 (required) | Rule 12.8 (required) |
| Rule 12.3 (advisory) | Rule 12.10 (required) |
| Rule 12.4 (advisory) | Rule 12.11 (advisory) |
| Rule 13.1 (required) | New |
| Rule 13.2 (required) | Rule 12.2 (required) |
| Rule 13.3 (advisory) | Rule 12.13 (advisory) |
| Rule 13.4 (advisory) | Rule 13.1 (required) |
| Rule 13.5 (required) | Rule 12.4 (required) |
| Rule 13.6 (mandatory) | Rule 12.3 (required) |
| Rule 14.1 (required) | Rule 13.4 (required) |
| Rule 14.2 (required) | Rule 13.5 (required) |
| | Rule 13.6 (required) |
| Rule 14.3 (required) | Rule 13.7 (required) |
| Rule 14.4 (required) | Rule 13.2 (advisory) |
| Rule 15.1 (advisory) | Rule 14.4 (required) |
| Rule 15.2 (required) | Rule 14.4 (required) |
| Rule 15.3 (required) | Rule 14.4 (required) |
| Rule 15.4 (advisory) | Rule 14.6 (required) |
| Rule 15.5 (advisory) | Rule 14.7 (required) |
| Rule 15.6 (required) | Rule 14.3 (required) |
| | Rule 14.8 (required) |
| | Rule 14.9 (required) |
| Rule 15.7 (required) | Rule 14.10 (required) |
| Rule 16.1 (required) | Rule 15.0 (required) |
| Rule 16.2 (required) | Rule 15.1 (required) |
| Rule 16.3 (required) | Rule 15.2 (required) |
| Rule 16.4 (required) | Rule 15.3 (required) |
| Rule 16.5 (required) | Rule 15.3 (required) |
| Rule 16.6 (required) | Rule 15.5 (required) |
| Rule 16.7 (required) | Rule 15.4 (required) |

| MISRA C:2012 | MISRA C:2004 |
|---|---|
| Rule 17.1 (required) | Rule 16.1 (required) |
| Rule 17.2 (required) | Rule 16.2 (required) |
| Rule 17.3 (mandatory) | Rule 8.1 (required) |
| | Rule 16.6 (required) |
| Rule 17.4 (mandatory) | Rule 16.8 (required) |
| Rule 17.5 (advisory) | New |
| Rule 17.6 (mandatory) | New |
| Rule 17.7 (required) | New |
| Rule 17.8 (advisory) | New |
| Rule 18.1 (required) | Rule 17.1 (required) |
| Rule 18.2 (required) | Rule 17.2 (required) |
| Rule 18.3 (required) | Rule 17.3 (required) |
| Rule 18.4 (advisory) | Rule 17.4 (required) |
| Rule 18.5 (advisory) | Rule 17.5 (advisory) |
| Rule 18.6 (required) | Rule 17.6 (required) |
| Rule 18.7 (required) | New |
| Rule 18.8 (required) | New |
| Rule 19.1 (mandatory) | Rule 18.2 (required) |
| Rule 19.2 (advisory) | Rule 18.4 (required) |
| Rule 20.1 (advisory) | Rule 19.1 (advisory) |
| Rule 20.2 (required) | Rule 19.2 (advisory) |
| Rule 20.3 (required) | Rule 19.3 (required) |
| Rule 20.4 (required) | Rule 19.4 (required) |
| Rule 20.5 (advisory) | Rule 19.6 (required) |
| Rule 20.6 (required) | Rule 19.9 (required) |
| Rule 20.7 (required) | Rule 19.10 (required) |
| Rule 20.8 (required) | New |
| Rule 20.9 (required) | Rule 19.11 (required) |
| Rule 20.10 (advisory) | Rule 19.13 (advisory) |
| Rule 20.11 (required) | Rule 19.12 (required) |
| Rule 20.12 (required) | New |
| Rule 20.13 (required) | Rule 19.16 (required) |
| Rule 20.14 (required) | Rule 19.17 (required) |
| Rule 21.1 (required) | Rule 20.1 (required) |
| Rule 21.2 (required) | Rule 20.2 (required) |
| Rule 21.3 (required) | Rule 20.4 (required) |
| Rule 21.4 (required) | Rule 20.7 (required) |
| Rule 21.5 (required) | Rule 20.8 (required) |

| MISRA C:2012 | MISRA C:2004 |
|---|---|
| Rule 21.6 (required) | Rule 20.9 (required) |
| Rule 21.7 (required) | Rule 20.10 (required) |
| Rule 21.8 (required) | Rule 20.11 (required) |
| Rule 21.9 (required) | New |
| Rule 21.10 (required) | Rule 20.12 (required) |
| Rule 21.11 (required) | New |
| Rule 21.12 (advisory) | New |
| Rule 22.1 (required) | New |
| Rule 22.2 (mandatory) | New |
| Rule 22.3 (required) | New |
| Rule 22.4 (mandatory) | New |
| Rule 22.5 (mandatory) | New |
| Rule 22.6 (mandatory) | New |