
Release Notes for the MISRA-C:2004 Compliance Module M2CM-3.2-QAC-8.1

July 2012

Contents

[Overview](#)

[Implementation Changes](#)

Unless otherwise noted, all materials provided in this release are Copyright (c) 2012 Programming Research.

All usage of this software must be in accordance with the terms of a current licence agreement with Programming Research as supplied with QA C.

Overview

This release of M2CM suppresses non MISRA messages in level 3 by default. This is reflected in the standard M2CM message personality and the User Guide.

New messages which have been introduced in the QAC parser (replacing messages marked as obsolete) have been included in this version of M2CM. Please refer to the QAC Release Notes for full details.

Implementation Changes

Rule	Rule Text	Notes
1.1	All code shall conform to ISO/IEC 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/COR2:1996192	<p>Added messages: 0410, 0609, 0611, 0612, 0614, 0639, 0647, 0715, 0739, 0828, 0857, 0858, 0859, 0875</p> <p>Added messages: 1045</p> <p>Added messages: 1046, 1047, 1048</p> <p>Added messages: 1055, 3236, 3237, 3238, 3244</p> <p>Added messages: 1056</p> <p>Messages associated with ISO:C90 environmental limits have been added to the enforcement of this rule. These messages were incorrectly omitted from previous releases of M2CM. (CR14648)</p> <p>Message1045 has been introduced to support the use of the #include_next language extension. (CR15552)</p> <p>Messages have been introduced to support the use of default argument syntax. (CR15552)</p> <p>Messages have been introduced to support the use of inline functions. (CR15552)</p> <p>Messages have been introduced to support the use of type _Bool. (CR15552)</p>
1.2	No reliance shall be placed on undefined or unspecified behaviour.	<p>Added messages: 3239</p> <p>Messages have been introduced to support the use of inline functions. (CR15552)</p>
3.1	All usage of implementation-defined behaviour shall be documented.	<p>Removed messages: 3702, 3703, 3704, 3705, 3706, 3707, 3832, 3833, 3902, 3903, 3904, 3905, 3906, 3907, 4032, 4033</p> <p>A number of messages associated with implementation defined behaviour of 'character types' are now classified as obsolete and have been removed from the enforcement of this rule. The same issues are addressed by new messages which are mapped to Rule 10.1. (CR15552)</p>
6.1	The plain char type shall be used only for the storage and use of character values.	<p>Removed messages: 3711, 3722, 3733, 3744, 3755, 3766, 3777, 3788, 3850, 3863,</p> <p>A number of messages associated with type conversions involving type 'char' are now classified as obsolete and have been removed from the enforcement of this rule. (CR15552)</p>

3911, 3922, 3933,
3944, 3955, 3966,
3977, 3988, 4050,
4063

Added messages:
1810, 1811, 1812,
1813, 4401, 4421,
4431, 4441, 4451,
4510, 4511, 4512,
4513, 4514, 4517,
4518, 4519

These messages have been introduced to provide improved enforcement by correctly identifying data which is 'essentially character' in nature although not necessarily of type 'char'. (CR15552)

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

Removed
messages: 3700,
3701, 3900, 3901

A number of messages associated with type conversions involving type 'char' are now classified as obsolete and have been removed from the enforcement of this rule. (CR15552)

Added messages:
4413, 4414

These messages have been introduced to provide improved enforcement by correctly identifying data which is 'essentially character' in nature although not necessarily of type 'char'. (CR15552)

10.1 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

Removed
messages: 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,
3908, 3909, 3910,
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,

Many of the messages associated with implicit type conversions are now classified as obsolete and have been removed from enforcement of this rule. (CR15552)

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, 4240,
 4241, 4242, 4243,
 4244, 4245.

Added messages:
 0570, 0572, 1256,
 1257, 1290, 1291,
 1292, 1293, 1294,
 1295, 1296, 1297,
 1298, 1299, 1800,
 1802, 1803, 1804,
 1810, 1811, 1812,
 1813, 1820, 1821,
 1822, 1823, 1824,
 1830, 1831, 1832,
 1833, 1834, 1840,
 1841, 1842, 1843,
 1844, 1850, 1851,
 1852, 1853, 1854,
 1860, 1861, 1862,
 1863, 1864, 1880,
 1881, 1882, 1890,
 1891, 4117, 4401,
 4402, 4403, 4404,
 4405, 4410, 4412,
 4413, 4414, 4415,
 4420, 4421, 4422,
 4423, 4424, 4425,
 4430, 4431, 4432,
 4434, 4435, 4436,
 4437, 4440, 4441,
 4442, 4443, 4445,
 4446, 4447, 4451,
 4460, 4461, 4463,
 4464, 4470, 4471,
 4480, 4481, 4490,
 4491, 4500, 4501,
 4502, 4503, 4504,
 4505, 4507.

Removed
 messages: 1401,
 1402, 1403, 1411,
 1412, 1413, 1421,
 1422, 1423, 1431,
 1433, 1441, 1442,
 1443

Removed
 messages: 3799,
 3800, 3801, 3802,
 3803, 3804, 3805,
 3806, 3807, 3810,
 3811, 3812, 3813,

These messages have been introduced to provide improved enforcement by correctly identifying the essential type (underlying type) of data and providing support for type `_Bool`. (CR15552)

These messages associated with enum usage are now classified as obsolete and have been superseded by messages 442x, 443x, 444x, 1296, 1297 etc. (CR15552)

Many of the messages associated with implicit type conversions are now classified as obsolete and have been removed from enforcement of this rule. (CR14459, CR15552)

10.2 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

		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 Added messages: 1264, 1265, 1266, 1880, 1881, 1882, 1892, 1893, 1894, 1895, 4450, 4452, 4453, 4454, 4462, 4465, 4472, 4482, 4492	These messages have been introduced to provide a more accurate enforcement of the rule using a smaller and better designed library of messages. (CR15552)
10.3	The value of a complex expression of integer type shall only be cast to a type of the same signedness that is no wider than the underlying type of the expression.	Removed messages: 4121 Added messages: 4390, 4391, 4393, 4394 Added messages: 4118	Message 4121 is now classified as obsolete. It fails to identify widening of complex expressions consistently. (CR15552) New messages have been introduced which provide more accurate identification of complex expressions using underlying type principles. (CR15552) Message 4118 is now mapped to this rule to identify the specific dangers of casting the result of an integer division to a floating type. (CR14459)
10.4	The value of a complex expression of floating type shall only be cast to a floating type that is narrower or of the same size.	Removed messages: 4126, 4127, 4128 Added messages: 4392, 4395	Messages 4126, 4127, 4128 are now classified as obsolete. They fail to identify widening of complex expressions consistently. (CR15552) New messages have been introduced which provide more accurate identification of complex expressions. (CR15552)
10.5	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.	Added messages: 4397, 4398, 4399, 4498, 4499	New messages have been introduced to provide enforcement of this rule based on underlying type principles. (CR15552)
11.3	A cast should not be performed between a pointer type and an integral type.	Added messages: 0360, 0361, 0362	New messages have been introduced which provide support for type _Bool. (CR15552)
12.7	Bitwise operators shall not be applied to operands whose underlying type is signed.	Removed messages: 0502, 4130, 4131 Added messages: 4532, 4533	Messages 502, 4130 and 4131 are now classified as obsolete. (CR15552) These messages have been introduced to provide more accurate enforcement. (CR15552)
13.2	Tests of a value against zero should be made explicit, unless the operand is effectively Boolean.	Added messages: 4528, 4529, 4538, 4539, 4548, 4549, 4558, 4559, 4568, 4569	New messages have been introduced to provide enforcement of this rule based on comprehensive identification of 'essentially Boolean' data based on underlying type principles. (CR15552)
14.1	There shall be no unreachable code.	Added messages: 2742	Message 2742 has been added in order to specifically identify the reason for unreachable code in an 'if' statement. (CR15552)
16.1	Functions shall not be defined with a variable	Removed	Message 5069, generated in secondary analysis,

	number of arguments.	messages: 5069	provided enforcement of this rule in previous versions of M2CM. It has been superseded by a new primary analysis message. (CR15552)
		Added messages: 1337	A new message has been introduced in primary analysis to enforce this rule. (CR15552)
16.4	The identifiers used in the declaration and definition of a function shall be identical.	Removed messages: 1334	Message 1334 has been removed from the enforcement to conform with a clarified interpretation of this MISRA rule. (CR15552)
16.8	All exit paths from a function with non-void return type shall have an explicit return statement with an expression.	Removed messages: 2877	Message 2877 is now more accurately mapped to Rule 21.1. (CR14612)
18.1	All structure and union types shall be complete at the end of a translation unit.	Removed messages: 3313	Message 3313 has been removed from the enforcement to conform with a clarified interpretation of this MISRA rule. (CR14459)
21.1	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	Added messages: 2877 Added messages: 2845, 2846, 2847, 2984, 2985, 2986	Message 2877 is now mapped to this rule instead of Rule 16.8. (CR14612) New dataflow messages have been introduced in QAC 8.1 which add to the enforcement of this rule. (CR15552)

PRQA and the PRQA logo are registered trademarks of Programming Research.

QA C, QA C++, QA Fortran, and QA J are trademarks of Programming Research.

Microsoft and Windows are Registered trademarks of Microsoft Corporation.

All other products or services mentioned in this document are covered by the trademarks, service marks, or product names as designated by the companies who market those products or services.