
User Guide for the MISRA-C:2004 Compliance Module M2CM-3.2-QAC-8.1.2

July 2013

Contents

[Introduction](#)
[MISRA-C](#)
[The MISRA-C:2004 Compliance Module](#)
[Installation](#)
[Documentation](#)
[M2CM Components](#)
[The Compiler Personality](#)
[The Analyser Personality](#)
[The Message Personality](#)
[The Dataflow Personality](#)
[Using M2CM](#)
[Rules and Messages](#)
[The QA C Message System](#)
[Mapping Messages to MISRA Rules](#)
[Global Checks](#)
[Command Line Usage](#)

Introduction

MISRA-C

The original MISRA C Guidelines were published in April 1998 by the Motor Industry Research Association in a document entitled "Guidelines For The Use Of The C Language In Vehicle Based Software". In 2004, an updated version of this document was published under the title "MISRA C:2004 Guidelines for the use of the C language in critical systems". In July 2007 a Technical Corrigendum was issued to address a number of issues which required correction or clarification. This was incorporated into the base document when MISRA C2 was reprinted in June 2008. Most recently in 2013, an updated version of the MISRA C Guidelines was published under the title "MISRA C:2012 Guidelines for the use of the C language in critical systems". There are therefore three versions of the guidelines and these are now referred to as MISRA C:1998 (MISRA C1), MISRA C:2004 (MISRA C2) and MISRA C:2012 (MISRA C3).

Although derived from MISRA-C:1998, MISRA-C:2004 contains a number of additional requirements and a few significant changes. Code that is compliant with MISRA-C:1998 will not necessarily be compliant with MISRA-C:2004 and code that is compliant with MISRA-C:2004 will not necessarily be compliant with MISRA-C:1998.

The MISRA-C:2004 Compliance Module

This document provides an introduction to the MISRA-C:2004 Compliance Module (M2CM). The M2CM module is designed to enforce compliance with the MISRA-C:2004 programming standard in conjunction with QA C. Please note that this version of M2CM is designed to operate with QA C Version 8.1.2 and will not function correctly with earlier releases.

QA C is a software tool that performs static analysis on C source code. Invariably a process of configuration is required in order to introduce QA C effectively to a particular software development environment. It is important to have QA C configured so as to restrict the output of warning messages to those that are of particular concern. M2CM is one such configuration, it configures QA C to enforce compliance with the MISRA-C:2004 coding rules.

QA C is able to detect most of the statically detectable conditions identified in the MISRA-C:2004 rules (as well as many others). M2CM configures QA C to identify issues which are specific to those rules, and provides a cross-reference between the standard QA C warning message(s) and the corresponding MISRA-C:2004 rule. Also available is an extended HTML description of each rule.

For some MISRA-C:2004 coding standard guidelines which are not enforceable directly by QA C, additional checks are provided

through a secondary-analysis program. These checks are programmed using the analysis output files (.met files) generated by the QA C "primary analysis" process.

Some MISRA-C:2004 rules, as noted in Paragraph 5.6 of the guidelines, impose constraints on the complete project rather than the code in a single translation unit. These requirements are enforced by the Cross-Module Analysis capability of QA C.

Installation

For details of installation please refer to the Installation Notes appropriate to your platform (e.g. M2CM-3.2-QAC-8.1.2-Win-Installation_Notes.pdf).

Documentation

When you have completed installation of M2CM-3.2-QAC-8.1.2, please refer to the following documents located in [m2cm/doc](#), which further describe the enforcement provided by MISRA-C:2004.

- M2CM-3.2-QAC-8.1.2-Coding_Standard_Manual
- M2CM-3.2-QAC-8.1.2-Coding_Standard_Review
- M2CM-3.2-QAC-8.1.2-*platform*-Installation_Notes
- M2CM-3.2-QAC-8.1.2-QAC_Message_Usage_Summary
- M2CM-3.2-QAC-8.1.2-Release_Notes
- M2CM-3.2-QAC-8.1.2-Rule_Enforcement
- M2CM-3.2-QAC-8.1.2-Rule_Xreference

M2CM Components

The M2CM compliance module introduces a number of additional components to your QA C installation which provide the following features:

1. A subset of the QA C messages appropriate to enforcement of the MISRA-C:2004 rules.
2. A new message structure reflecting the MISRA-C:2004 rules. This is defined by the user message file qac usr.m2cm.
3. Secondary-analysis checks to enforce a number of the MISRA-C:2004 rules.

A set of default personalities is supplied as described below. It is good practice to use a copy of these personalities rather than using the original files. When changing personalities throughout an existing project with multiple folders, remember to apply the new personality to the root folder and then propagate the change to all sub-folders.

The Compiler Personality

M2CM is supplied with a default compiler personality in the personalities directory ([m2cm/personalities/m2cm.p_c](#)). This contains the following option settings configured in the "Identifiers" tab:

Internal Identifier Significant Length:	31
External Identifier Significant Length:	31
Ignore External Identifier Case:	Disabled

This compiler personality may be introduced directly into a project and modified as required. However, it may be more convenient to merge the above settings into another existing compiler personality.

The Analyser Personality

M2CM is supplied with a default analyser personality in the personalities directory ([m2cm/personalities/m2cm.p_a](#)). This contains the following option settings:

1. In the "K&R Compatibility" tab, the options for "Using the glue operator '##'" and "Using the stringify operator '#'" are enabled.
2. In the "Warning Calls" tab, a number of warning call messages are configured.

This analyser personality may be introduced directly into a project and modified as required. However, it may be more convenient to merge the above settings into another existing analyser personality.

The Message Personality

M2CM is supplied with a default message personality in the personalities directory ([m2cm/personalities/m2cm.p_s](#)).

The message personality is a fundamental component of M2CM. It configures the following:

1. The set of messages appropriate to enforcement of the MISRA rules.

2. The name and location of a user message file are specified using the -UP and -USR options, ([m2cm/messages/qac usr.m2cm](#)) which provides a mapping of messages to the appropriate MISRA rule. The location of a user message file is treated by QA C as the first location in which to search for supporting HTML (before searching the QA C messages directory).
3. Configuration of secondary-analysis. The specification of the secondary-analysis program, script file and parameters is recorded in the message personality in the form of comments. It is therefore possible to switch between M2CM and non-M2CM configurations by simply selecting a different message personality.

The Dataflow Personality

M2CM is supplied with a default dataflow personality in the personalities directory ([m2cm/personalities/m2cm.p_d](#)).

Dataflow analysis is required for the enforcement of a number of MISRA rules. The supplied dataflow personality enables dataflow and applies some typical dataflow configuration options. An appropriate configuration for dataflow analysis will depend on factors such as the hardware environment and the complexity of the code. These issues are discussed in the dataflow reference manual. There are no configuration requirements specific to M2CM enforcement and there is therefore no particular need to use the supplied file `m2cm.p_d` rather than any other dataflow personality file.

Using M2CM

Rules and Messages

MISRA-C:2004 defines 142 rules grouped in 21 sections (e.g. Environment, Language Extensions, Documentation, Character Sets, Identifiers). Each rule is classified as either Required or Advisory.

QA C messages are arranged in 10 levels. In M2CM-3.2-QAC-8.1.2, just 6 of these levels are used, as follows:

Level 0: Information

Level 1: Obsolete

Level 2: Minor

Level 3: Major

Level 4: M2CM

Level 9: Errors

Level 0 contains informational messages, sub-messages and error recovery messages.

Messages in Level 1 are classified as obsolete for a variety of reasons, but usually because their functionality has been superseded by messages which are functionally superior or better specified. Obsolete messages are retained in the tool for the convenience of users of previous versions of QA C and are not used in this version of M2CM.

Messages in Levels 2 and 3 are not directly relevant to MISRA-C compliance. These messages are disabled by default but may be enabled at the user's discretion. It should be noted that many of them identify code conditions which can be of significant interest or concern.

QA C 8.1.2 includes a large number of new messages which are designed to supersede existing messages particularly in the area of arithmetic type conversions. These new messages have been implemented in order to provide improved enforcement of MISRA-C:2004 and in order to reduce the incidence of false positive messages. In order to provide continuity to users of previous versions of M2CM, this version does not make use of most of these new messages.

Level 4 contains a message group corresponding to each MISRA-C:2004 rule. A rule may be enforced by QA C (primary analysis), Dataflow analysis, CMA (Cross-Module Analysis), Secondary-Analysis or by a combination of these 4 elements.

Level 9 contains configuration errors and syntax errors. These are "hard errors" which cannot be disabled or suppressed; they should be resolved before meaningful analysis of the code can be conducted. The results of any analysis in which level 9 messages are generated must be treated as unreliable.

Mapping Messages to MISRA Rules

Some analysis messages are associated with more than one MISRA Rule. This occurs when:

- The requirements of 2 MISRA rules overlap. For example, any violation of Rule 16.6 ("The number of arguments passed to a function shall match the number of parameters") will also violate either Rule 1.1 or Rule 1.2., or
- The granularity of the condition identified by an analysis message does not coincide uniquely with a single MISRA Rule. For example, message 2212 ("Body of control statement is not enclosed within braces.") is mapped to both Rule 14.8 (The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement) and Rule 14.9 (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).

The rule or rules with which each message is associated will always be reported in the "reference text". However, in the situations

where a message is associated with more than one rule, it will only be specifically *mapped* to one of the rules - the *primary* rule. Where this situation occurs, any statistics relating to the number of violations of any particular rule may be slightly misleading.

It is also important to note that if a message which is mapped to more than one rule is disabled, it will be disabled for all the rules with which it is associated.

Global Checks

M2CM-3.2-QAC-8.1.2 implements global checks using the QA C 8.1.2 Cross-Module Analysis capability. This analysis is normally only of significant value when the code from a complete project is available for analysis.

For details of configuration and use of QA C Cross-Module Analysis, please refer to the QA C User Guide.

Command Line Usage

Within the M2CM package, the `m2cm/bin` directory contains file(s) for configuring the command line environment (`m2cmconf.bat` on Windows and `.profile` and `.cshrc` for Unix). These set the standard QA C environment variables, in particular QACPATH and QACBIN. The environment variable CMBIN which locates the compliance module bin directory is also set for backward compatibility, but is not used by the M2CM-3.2-QAC-8.1.2 package.

Command Line analysis and viewing is facilitated by the separately available QAW product. Please refer to www.programmingresearch.com/downloads for this and to the included user guide for details of usage. As an example, qaw allows project analysis as follows:

```
qaw qac m2cm_examples.prj -view
```

It is also possible to perform analysis based on a Makefile. For details of this refer to MakefileIntegration at www.programmingresearch.com/downloads.

TRADEMARKS

MISRA is a registered trademark of MIRA Limited, held on behalf of the MISRA Consortium.

IMPORTANT NOTICE

DISCLAIMER OF WARRANTY

The staff of Programming Research Group have taken due care in preparing this document which is believed to be accurate at the time of publication. However, no liability can be accepted for errors or omissions nor should this document be considered as an expressed or implied warranty that the products described perform as specified within.

COPYRIGHT NOTICE

This document is copyrighted Copyright (c) Programming Research 2013, and may not, in whole or in part, be copied, reproduced, disclosed, transferred, translated, or reduced to any form, including electronic medium or machine-readable form, or transmitted by any means, electronic or otherwise, unless Programming Research Group consents in writing in advance.

TRADEMARKS

PRQA, the PRQA logo, QA C++ and QA C are registered trademarks of Programming Research Group. Windows is a registered trademark 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.

CONTACTING PROGRAMMING RESEARCH

For technical support, contact your nearest Programming Research Group authorised distributor or you can contact Programming Research's head office:

by telephone on +44 (0) 1 932 888 080

by fax on +44 (0) 1 932 888 081

or by e-mail on support@programmingresearch.com

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.