
User Guide for the MISRA C:2012 Compliance Module M3CM-1.0-QAC-8.1.2

July 2013

Contents

[Introduction](#)

[MISRA C](#)

[The MISRA C:2012 Compliance Module](#)

[Installation](#)

[Documentation](#)

[M3CM Components](#)

[The Compiler Personality](#)

[The Analyser Personality](#)

[The Message Personality](#)

[The Dataflow Personality](#)

[Using M3CM](#)

[Directives and Rules](#)

[Configuring M3CM for conformance to C90 or C99](#)

[The QA C Message System](#)

[Mapping Messages to MISRA Directives and Rules](#)

[GlobalChecks](#)

[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 C2, MISRA C3 contains some significant changes. The Guidelines are divided into Directives and Rules, and each is classified as either Advisory, Required, or Mandatory. Code that is compliant with MISRA C2 will not necessarily be compliant with MISRA C3 and code that is compliant with MISRA C3 will not necessarily be compliant with MISRA C2.

The MISRA C:2012 Compliance Module

This document provides an introduction to the MISRA C:2012 Compliance Module (M3CM). The M3CM module is designed to enforce compliance with the MISRA C:2012 programming standard in conjunction with QA C. Please note that this version of M3CM 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. M3CM is one such configuration, it configures QA C to enforce compliance with the MISRA C:2012 coding guidelines.

QA C is able to detect most of the statically detectable conditions identified in the MISRA C:2012 guidelines (as well as many others). M3CM configures QA C to identify issues which are specific to those guidelines, and provides a cross-reference between the standard QA C warning message(s) and the corresponding MISRA C:2012 guideline. Also available is an HTML description of each guideline.

For some MISRA C:2012 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:2012 rules, as noted in Section 6.6 of the guidelines, impose constraints on the complete system 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. M3CM-1.0-QAC-8.1.2-Win-Installation_Notes.pdf).

Documentation

When you have completed installation of M3CM-1.0-QAC-8.1.2, please refer to the following documents located in [m3cm/doc](#), which further describe the enforcement provided by MISRA C:2012.

- M3CM-1.0-QAC-8.1.2-Coding_Standard_Manual
- M3CM-1.0-QAC-8.1.2-Coding_Standard_Review
- M3CM-1.0-QAC-8.1.2-*platform*-Installation_Notes
- M3CM-1.0-QAC-8.1.2-QAC_Message_Usage_Summary
- M3CM-1.0-QAC-8.1.2-Release_Notes
- M3CM-1.0-QAC-8.1.2-Rule_Enforcement
- M3CM-1.0-QAC-8.1.2-Rule_Xreference

M3CM Components

The M3CM 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:2012 guidelines.
2. A new message structure reflecting the MISRA C:2012 guidelines. This is defined by the user message file `qac usr.m3cm`.
3. Secondary-analysis checks to enforce a number of the MISRA C:2012 guidelines.

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

M3CM is supplied with a default compiler personality in the personalities directory ([m3cm/personalities/m3cm.p_c](#)). This contains the following configuration option settings:

"Identifiers" tab:

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

"Extensions" tab:

Extension class ASM	Enabled
Extension class long long	Enabled

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

M3CM is supplied with a default analyser personality in the personalities directory ([m3cm/personalities/m3cm.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

M3CM is supplied with a default message personality in the personalities directory ([m3cm/personalities/m3cm.p_s](#)).

The message personality is a fundamental component of M3CM. It configures the following:

1. The set of messages appropriate to enforcement of the MISRA guidelines.
2. The name and location of a user message file are specified using the -UP and -USR options, ([m3cm/messages/qac_usr.m3cm](#)) which provides a mapping of messages to the appropriate MISRA guideline. 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 M3CM and non-M3CM configurations by simply selecting a different message personality.

The Dataflow Personality

M3CM is supplied with a default dataflow personality in the personalities directory ([m3cm/personalities/m3cm.p_d](#)).

Dataflow analysis is required for the enforcement of a number of MISRA guidelines. 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 M3CM enforcement and there is therefore no particular need to use the supplied file `m3cm.p_d` rather than any other dataflow personality file.

Using M3CM

Directives and Rules

MISRA C:2012 defines 159 guidelines consisting of 16 directives and 143 rules. The guidelines are grouped into a number of sections (e.g. Compilation and build, Code design, Unused code, Types, Pointer type conversions).

The MISRA C:2012 document describes the distinction between rules and directives as follows:

"A directive is a guideline for which it is not possible to provide the full description necessary to perform a check for compliance. Additional information, such as might be provided in design documents or requirements specifications, is required in order to be able to perform the check. Static analysis tools may be able to assist in checking compliance with directives but tools may place widely different interpretations on what constitutes a non-compliance."

"A rule is a guideline for which a complete description of the requirement has been provided. It should be possible to check that source code complies with a rule without needing any other information. In particular, static analysis tools should be capable of checking compliance with rules subject to the limitations described in Section 6.5."

M3CM describes directives as "Not Statically Enforceable (NSE)". Enforcement is provided for some directives but this may not be sufficient by itself to ensure compliance in every situation. It should be noted that the percentage of enforcement indicated by the Rule Enforcement report excludes all directives even though partial enforcement is available for some of them.

Configuring M3CM for conformance to C90 or C99

MISRA C:2012 rules are classified as being related to the C90 Language Standard, the C99 Language Standard, or both (as described in Appendix F). By default, M3CM is configured to produce messages for systems conforming to the C90 standard and in this configuration, diagnostic messages will be triggered wherever a C99 feature is used.

If M3CM is to be used in a system conforming to C99 (or a subset of the additional features in C99), a number of messages may be disabled.

1. QA C 8.1 has 19 messages which identify the use of C99 features: 0180, 0320, 0604, 0617, 0618, 0850, 0930, 0945, 1011, 1018, 1027, 1030, 1031, 1051, 1052, 1053, 1054, 1055, 1056. All these messages are prefixed with "[C99]":
2. QA C 8.1 has 19 messages which identify a breach of a C90 "environmental limit": 0410, 0609, 0611, 0612, 0614, 0639, 0647, 0715, 0739, 0776, 0778, 0810, 0815, 0816, 0828, 0857, 0858, 0859, 0875. These are messages which identify potential limits on portability which apply in C90 but not in a C99 environment. All these messages contain the text "ISO:C90".

The QA C Message System

Each MISRA C Guideline is classified as either Mandatory, Required or Advisory and this classification is reflected in the structure of the message system. QA C messages are arranged in 10 levels, which are configured in M3CM-1.0-QAC-8.1.2 as follows:

Level 0: Information

Level 1: **Obsolete**
Level 2: **Minor**
Level 3: **Major**
Level 4: **MISRA Advisory Directives**
Level 5: **MISRA Advisory Rules**
Level 6: **MISRA Required Directives**
Level 7: **MISRA Required Rules**
Level 8: **MISRA Mandatory Rules**
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 M3CM.

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.

Levels 4-8 contain message groups corresponding to each MISRA C:2012 guideline, divided into advisory, required and mandatory rules and directives. This separation allows users to disable, for example, all advisory items in order to focus only on required and mandatory guidelines.

A rule or directive may be enforced by

1. QA C (primary analysis)
2. Dataflow analysis
3. Secondary Analysis
4. CMA (Cross-Module Analysis)
5. by some combination of these 4 elements.

The type of analysis used to enforce a given guideline is listed in the Cross Reference Listing.

Level 9 contains configuration errors and syntax errors. These are "hard errors" which cannot be disabled or suppressed; they should always 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 Directives and Rules

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

- The requirements of 2 MISRA guidelines overlap. For example, any violation of Rule 11.7 ("A cast shall not be performed between pointer to object and a non-integer arithmetic type") will also violate Rule 1.3, or
- The granularity of the condition identified by an analysis message does not coincide uniquely with a single MISRA Guideline. For example, message 2771 ("Definite: These pointers address different objects.") is mapped to both Rule 18.2 ("Subtraction between pointers shall only be applied to pointers that address elements of the same array") and Rule 18.3 ("The relational operators <, <=, > and >= shall not be applied to objects of pointer type except where they point into the same object"), or
- Violation of one rule may also result in violation of another. For example Rule 5.8 ("Identifiers that define objects or functions with external linkage shall be unique") and Rule 5.9 ("Identifiers that define objects or functions with internal linkage should be unique").

The guideline or guidelines with which each message is associated will always be reported in the "reference text". However, in the (few) situations where a message is associated with more than one guideline, it will only be specifically *mapped* to one of the guidelines - the *primary* guideline. Where this situation occurs, any statistics relating to the number of violations of any particular guideline may be slightly misleading.

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

Global Checks

M3CM-1.0-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 M3CM package, the `m3cm/bin` directory contains file(s) for configuring the command line environment (`m3cmconf.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 M3CM-1.0-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 m3cm_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.