MISRA C:2023 Addendum 4

Coverage of MISRA C:2023 against ISO/IEC TR 24472 “Language vulnerabilities”

October 2024

First published October 2024 by The MISRA Consortium Limited
1 St James Court
Whitefriars
Norwich
Norfolk
NR3 1RU
UK

www.misra.org.uk

ISBN 978-1-911700-16-6 PDF

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

MISRA C:2023 Addendum 4

Coverage of MISRA C:2023 against ISO/IEC TR 24472 “Language vulnerabilities”

October 2024

MISRA Mission Statement

MISRA provides world-leading best practice guidelines for the safe and secure application of both embedded control systems and standalone software.

MISRA is a collaboration between manufacturers, component suppliers, engineering consultancies and academics which seeks to research and promote best practice in developing safety- and security-related electronic systems and other software-intensive applications.

To this end, MISRA conducts research projects and publishes documents that provide accessible information for engineers and management.

MISRA also facilitates the dissemination and exchange of information between practitioners through supporting and holding technical events.

Disclaimer

Adherence to the requirements of this document does not in itself ensure error-free robust software or guarantee portability and re-use.

Compliance with the requirements of this document, or any other standard, does not of itself confer immunity from legal obligations.

Foreword

It is a widely held viewpoint that MISRA C provides best-practice guidelines for the development of safety-related systems.

The MISRA C Working Group monitors for additional sources that may assist in the improvement of that guidance, particularly relating to possible language vulnerabilities, and has been monitoring the work of ISO/IEC JTC1/SC22/WG23 as it prepared the ISO/IEC 24772 *Guidance to avoiding vulnerabilities in programming languages* series of standards.

The publication by ISO/IEC JTC1/SC22/WG23 of the language-independent guidance of ISO/IEC 24772-1:2019 Part 1 [2], closely followed by the C language specific guidance of ISO/IEC 24772-3:2020 Part 3 [3] allowed the MISRA C Working Group to compile this Addendum, which documents the coverage of MISRA C against these two standards.

Andrew Banks FBCS CITP
Chairman, MISRA C Working Group

Acknowledgements

The MISRA C Working Group

The MISRA Consortium would like to thank the following members of the MISRA C Working Group for their significant contribution to the writing of this document:

Andrew Banks	LDRA Ltd (also Intuitive Consulting)
Jill Britton	Perforce
Clive Pygott	Columbus Computing Ltd

The MISRA Consortium also wishes to acknowledge contributions from the following members of the MISRA C Working Group during the development and review process:

Dave Banham	BlackBerry Ltd
Daniel Kästner	AbsInt Angewandte Informatik GmbH
Gerlinde Kettl	Vitesco Technologies GmbH
Chris Tapp	Keylevel Consultants Ltd

The MISRA Consortium Limited also wishes to acknowledge contributions from the following individuals during the development and review process:

David Ward	HORIBA MIRA Limited

Other acknowledgements

DokuWiki was used extensively during the drafting of this document. Our thanks go to all those involved in its development.

This document was typeset using fonts licensed under the SIL Open Font License, Version 1.1:

- Open Sans — Copyright © 2020, The Open Sans Project Authors.

Contents

1 Introduction

1.1 Background

Throughout the development of MISRA C, the main focus has been to address vulnerabilities in the C language, particularly for use in embedded systems, and primarily targeted at safety-related applications. One of the great successes of MISRA C has been its adoption across many industries, and in environments where safety-criticality is less of a concern, but where security is more of an issue.

The MISRA C Working Group has been monitoring the work of ISO/IEC JTC1/SC22/WG23 as it prepared the ISO/IEC 24772 *Guidance to avoiding vulnerabilities in programming languages* series of standards, and this addendum shows the mapping of MISRA C to those standards.

1.2 Glossary

In this document:

- *Language vulnerability* means a concern identified in ISO/IEC 24772-1:2019 *Part 1: Language-independent guidance* [2] and/or ISO/IEC 24772-3:2020 *Part 3: C* [3]
- *MISRA C* means MISRA C:2023 *Guidelines for the use of the C language in critical systems* [1]

1.3 Applicability

This document provides a mapping of the language vulnerabilities identified in ISO/IEC 24772-1:2019 *Part 1: Language-independent guidance* [2] and/or ISO/IEC 24772-3:2020 *Part 3: C* [3] against MISRA C.

This document should be read in conjunction with MISRA C:2023 *Guidelines for the use of the C language in critical systems* [1].

2 Coverage

2.1 Coverage classification

The coverage of each language vulnerability against MISRA C is classified as follows:

Status	Interpretation
Explicit	The language vulnerability is EXPLICITLY covered by one or more MISRA C guidelines, which directly addresses the undesired behaviour.
Implicit	The language vulnerability is IMPLICITLY covered by one or more MISRA C guidelines, although the behaviour is not explicitly referenced.
Restrictive	The language vulnerability is covered by one or more MISRA C guidelines that prohibit a language feature in a RESTRICTIVE manner.
Partial/Explicit	Some aspects of the language vulnerability are covered in a EXPLICIT manner. However, some aspects of the vulnerability are not covered by any MISRA C guideline.
None	The language vulnerability is not covered by any MISRA C guideline.
N/A	The language vulnerability is not applicable to the C language.

2.2 Coverage strength

The strength of the coverage of each language vulnerability against MISRA C is classified as follows:

Status	Interpretation
Strong	The language vulnerability by one or more targeted MISRA C Rules, (excluding Rule 1.3 on its own)
Weak	The language vulnerability is only covered by one or more MISRA C Directives, or by Rule 1.3.
None	The language vulnerability is not covered by any MISRA C Guidelines.

Note: For language vulnerabilities with Partial coverage, a combination of Strength coverages is shown.

3 ISO/IEC TR 24772 cross reference

3.1 Guideline By Guideline

3.1.1 ISO/IEC TR 24772-1 — Language independent

ISO/IEC TR 24772-1 Recommendations		MISRA C:2023 Guidelines			Comments
		Guidelines	Coverage		
5.01	HFC				See 6.11
5.02	HCB				See 6.08
5.03	STR				See 6.03
5.04	XYW				See 6.01
5.05	XYH				See 6.13
5.06	XYK				See 6.14
5.07	LAV				See 6.22
5.08	FIF				See 6.15
5.09	FIF				See 6.15
5.10	FLC				See 6.06

3.1.2 ISO/IEC TR 24772-3 — C Language

ISO/IEC TR 24772-1 Recommendations		MISRA C:2023 Guidelines			Comments
		Guidelines	Coverage		
6.01	-	-	N/A	None	General guidance
6.02	IHN	Dir 4.6 Rules 10.1-10.5	Explicit	Strong	Essential type model
6.03	STR	Rule 6.1, Rule 6.2, Rule 12.2	Explicit	Strong	
6.04	PLF	Dir 1.1, Dir 4.15 Rules 10.1-10.5, Rule 14.1	Explicit	Strong	
6.05	CCB	Rule 8.12	Explicit	Strong	
6.06	FLC	Rule 7.2, Rule 10.1, Rule 10.3, Rule 10.4, Rule 10.6, Rule 10.7, Rule 10.8, Rules 11.1-11.8	Explicit	Strong	Essential type model
6.07	CJM	Rule 21.16	Explicit	Strong	
6.08	HCB	Rule 18.1, Rule 21.17, R21.18	Explicit	Strong	
6.09	XYZ	Dir 4.1 Rule 18.1, Rule 21.7	Partial/ Explicit	Strong	No coverage of Appendix K
6.10	XYW	Rule 18.6, Rule 21.15, Rule 21.16, Rule 21.17	Explicit	Strong	
6.11	HFC	Dir 4.1 Rules 11.1-11.8	Explicit	Strong	
6.12	RVG	Dir 4.1 Rules 18.1-11.4	Explicit	Strong	
6.13	XYH	Dir 4.1, Dir 4.14, Rule 1.3	Explicit	Weak	
6.14	XYK	Dir 4.12 Rule 22.1, Rule 22.2	Implicit	Strong	
6.15	FIF	Dir 4.1 Rule 10.1, Rule 10.3, Rule 10.4, Rule 10.6, Rule 10.7, Rule 12.4	Explicit	Strong	

ISO/IEC TR 24772-1 Recommendations		MISRA C:2023 Guidelines			Comments
		Guidelines	Coverage		
6.16	PIK	Rule 10.1, Rule 10.3, Rule 10.4, Rule 10.6, Rule 10.7, Rule 12.2, Rule 12.4	Implicit	Strong	
6.17	NAI	Dir 4.5 Rule 1.1, Rule 5.1	Explicit	Strong	
6.18	WXQ	Rule 2.2	Explicit	Strong	
6.19	YZS	Rule 2.8	Explicit	Strong	
6.20	YOW	Dir 4.5 Rule 5.1, Rule 5.2, Rule 5.3, Rule 5.8, Rule 5.9, Rule 21.1, Rule 21.2	Explicit	Strong	
6.21	BJL	-	N/A	None	C has single namespace
6.22	LAV	Rule 9.1, Rule 9.2, Rule 9.3, Rule 9.7	Explicit	Strong	
6.23	JCW	Rule 10.1, Rule 12.1, Rule 13.2, Rule 14.4, Rule 20.7, Rule 20.10, Rule 20.11	Explicit	Strong	
6.24	SAM	Rule 12.1, Rule 13.2, Rule 13.5, Rule 13.6	Explicit	Strong	
6.25	KOA	Rule 2.2, Rule 13.4, Rule 14.3	Explicit	Strong	
6.26	XYQ	Dir 4.4 Rule 2.1, Rule 2.2	Explicit	Strong	
6.27	CLL	Rules 16.1-16.6	Explicit	Strong	
6.28	EOJ	Rule 15.6, Rule 15.7	Restrictive	Strong	Requires braces not additional else
6.29	TEX	Rule 14.1, Rule 14.2	Explicit	Strong	
6.30	XZH	Dir 4.1 Rule 1.3, Rule 18.1, Rule 21.6, Rule 21.17, Rule 21.18	Implicit	Strong	
6.31	EWD	Rule 15.1, Rule 15.2, Rule 15.3, Rule 15.5, Rule 21.4	Explicit	Strong	
6.32	CSJ	Dir 4.1, Dir 4.7 Rule 8.2, Rule 8.3, Rule 8.13, Rule 17.1, Rule 17.2, Rule 17.3	Explicit	Strong	
6.33	DCM	Dir 4.1 Rule 18.6, Rule 18.9	Explicit	Strong	
6.34	OTR	Rules 8.2-8.4, Rule 17.1, Rule 17.3	Explicit	Strong	
6.35	GDL	Rule 17.2	Explicit	Strong	
6.36	OYB	Dir 4.7	Explicit	Weak	
6.37	AMV	Rule 19.1, Rule 19.2	Explicit	Strong	
6.38	YAN	-	N/A	None	
6.39	XYL	Dir 4.12 Rule 22.1	Explicit	Strong	
6.40	SYM	-	N/A	None	C does not implement these mechanisms
6.41	RIP	-	N/A	None	C does not implement this mechanism
6.42	BLP	-	N/A	None	C does not implement polymorphism
6.43	PPH	-	N/A	None	C does not implement this mechanism
6.44	BKK	-	N/A	None	C does not implement this mechanism

ISO/IEC TR 24772-1 Recommendations		MISRA C:2023 Guidelines			Comments
		Guidelines	Coverage		
6.45	LRM	-	N/A	None	C does not implement this mechanism
6.46	TRJ	Dir 4.1, Dir 4.11 Rule 1.3, Rules 21.2-21.8, Rule 21.10, Rule 21.22, Rule 21.23	Explicit	Strong	
6.47	DJS	-	N/A	None	The C standard is silent on this
6.48	NYY	-	N/A	None	The C standard is silent on this
6.49	NSQ	-	N/A	None	The C standard is silent on this
6.50	HJW	-	N/A	None	C does not have exceptions and so cannot handle exceptions passed from other language systems
6.51	NMP	Dir 4.9 Rule 1.3, Rule 20.5, Rule 20.6	Explicit	Strong	
6.52	MXB	-	N/A	None	C does not implement run-time checking
6.53	SKL	Rule 1.1, Rule 1.3, Appendix H	Explicit	Strong	
6.54	BRS	Rule 1.1	Implicit	Strong	
6.55	BQF	Rule 1.3, Appendix H	Explicit	Weak	
6.56	EWF	Rule 1.1, Appendix H Rule 5.4, Rule 18.2, Rule 18.3, Rule 20.2	Explicit	Strong	
6.57	FAB	Dir 1.1, Rule 1.2, Appendix G Rule 5.4, Rule 18.2, Rule 18.3, Rule 20.2	Explicit	Strong	
6.58	MEM	Rule 1.5	Explicit	Strong	
6.59	CGA	Dir 4.7 Rule 17.7	Explicit	Strong	
6.60	CGT	Rule 22.15	Explicit	Strong	
6.61	CGX	Dir 5.1, Dir 5.2 Rule 9.7, Rule 12.6, Rule 22.14, Rules 22.16-19	Explicit	Strong	
6.62	CGS	Dir 5.2 Rule 22.16	Explicit	Strong	
6.63	CGM	Rule 22.12, Rule 22.14-19	Explicit	Strong	
6.64	SHL	Dir 4.11, Rule 1.3, Rule 21.6	Restrictive	Strong	

3.2 Coverage Summary

In summary, the coverage of MISRA C:2023 against the language vulnerabilities is as follows:

MISRA C Coverage		
Coverage	Strength	Number
Restrictive	Strong	2
	Weak	0
Explicit	Strong	40
	Weak	3
Implicit	Strong	4
	Weak	0
Partial/Explicit	Strong	1
	Weak	0
N/A	None	14
Total		64

4 References

4.1 MISRA C

[1] MISRA C:2023 *Guidelines for the use of the C language in critical systems*
ISBN 978-1-911700-08-1 (paperback), ISBN 978-1-911700-09-8 (PDF),
The MISRA Consortium Limited, Norwich, April 2023

4.2 ISO/IEC Standards

[2] ISO/IEC TR 24772-1:2019
Programming languages — Guidance to avoiding vulnerabilities in programming languages — Part 1: Language-independent guidance

[3] ISO/IEC TR 24772-3:2020
Programming languages — Guidance to avoiding vulnerabilities in programming languages — Part 3: C