MISRA C:2025 Addendum 6

Applicability of MISRA C:2025 to the Rust Programming Language

March 2025

First published March 2025 by The MISRA Consortium Limited
1 St James Court
Whitefriars
Norwich
Norfolk
NR3 1RU
UK

www.misra.org.uk

ISBN 978-1-911700-22-7

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

MISRA C:2025 Addendum 6

Applicability of MISRA C:2025 to the Rust Programming Language

March 2025

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

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 data-security is more of an issue.

With the emergence of the Rust programming language, there is a perception that all problems are solved simply by switching the programming language - but this is far from the case…

This document has been produced to assess the applicability of the MISRA C guidelines to the Rust programming language, and will form the basis of future work within MISRA, in collaboration with the wider Rust community.

Andrew Banks FBCS CITP
Chairman, MISRA C Working Group

Acknowledgements

The MISRA C Working Group (and Friends)

Note: An asterisk following a name denotes contributors who are not members of the MISRA C Working Group.

The MISRA Consortium would like to thank the following individuals, and their employers, for their significant contribution to the writing of this document:

	Andrew Banks	LDRA Ltd (also Intuitive Consulting)
	Roberto Bagnara	BUGSENG (and the University of Parma)
	Alex Celeste	Perforce
*	Nicola Vetrini	BUGSENG

The MISRA Consortium would like to thank the following individuals, and their employers, for their contribution during the review phase of this document:

	Jill Britton	Perforce
*	Luca Ciucci	BUGSENG
	Douglas Deslauriers	Vector Informatik GmbH
	Daniel Kästner	AbsInt Angewandte Informatik GmbH
	Gerlinde Kettl	Vitesco Technologies GmbH
	Gavin McCall	Codethink Ltd
	Chris Miller	GE Aerospace
*	Federico Serafini	BUGSENG
*	Daniel Silverstone	Codethink Ltd

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

Chris Tapp	Keylevel Consultants
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
- Fira Code — Copyright 2014–2020, The Fira Code Project Authors

Contents

1 Introduction 1

1.1 Background 1

1.2 Scope 1

2 Glossary 2

2.1 Applicability Rationale 2

2.2 Applicability Classification 2

2.3 Rust Categorization 2

3 Rust Cross Reference 3

4 References 12

1 Introduction

1.1 Background

This document has been produced to assess the applicability of the MISRA C guidelines to the Rust programming language, and will form the basis of future work within MISRA, in collaboration with the wider Rust community.

The intended purpose of this document is to highlight which areas of the Rust programming language are and are not subject to issues identified as affecting C projects.

By identifying which issues enforced by a given MISRA C guideline categorically cannot apply to a Rust project, developers of a Rust project can be certain that their code is compliant to whichever parts of functional safety standards those guidelines are intended to enforce. Where an issue enforced by a MISRA C guideline does still apply or potentially apply to code written in Rust, deeper analysis may still be required to ensure compliance, but the overall list is reduced and the relevant issues are clearly identified as mapping to some known enforcement which can aid in manual review.

1.2 Scope

This document provides an assessment of the applicability of the guidelines specified in MISRA C [1] to the Rust programming language [2] ("Rust"), not to the source code of any particular project written in Rust.

This document should be read in conjunction with MISRA C:2025 *Guidelines for the use of the C language in critical systems* [1].

2 Glossary

2.1 Applicability Rationale

The rationale for each MISRA C guideline is classified with one, or more, of the following:

Status	Interpretation
UB	The MISRA C guideline applies to C *Undefined* or *Unspecified Behaviour*
IDB	The MISRA C guideline applies to C *Implementation-defined Behaviour*
CQ	The MISRA C guideline applies to *Code Quality* considerations
DC	The MISRA C guideline applies to *Developer Confusion*, where there is common misunderstanding of a C feature

2.2 Applicability Classification

Applicability means that the identified undesirable behaviour is also present in Rust, and the mitigation in MISRA C is also relevant to Rust, but with a similar, not necesarily the same, solution.

Within the Assessment Matrix, there are two colums for *Applicability*:

1. Rust in general, including safe, *unsafe*, and foreign function interfaces; and
2. Safe Rust, excluding *unsafe* and *extern*.

The applicability of each MISRA C guideline to the Rust is classified as follows:

Status	Interpretation
Yes	The MISRA C guideline applies equally to Rust
No	The MISRA C guideline does not apply to Rust
Partial	The MISRA C guideline partially applies to some aspects of Rust

2.3 Rust Categorization

Each MISRA C guideline is assigned a category for Rust, as follows:

Status	Interpretation
Required	Code shall comply with this guideline, with a formal deviation required where this is not the case
Advisory	These are recommendations which should be followed as far as is reasonably practical. Formal deviation is not necessary for advisory guidelines but, if the formal deviation process is not followed, alternative arrangements should be made for documenting non-compliances
Disapplied	These are guidelines for which compliance is not required. No enforcement is expected, and any non-compliance may be disregarded
N/A	The behaviour does not apply to Rust

Note: the MISRA C category of *mandatory* is not used in this document.

3 Rust Cross Reference

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
Directives								
D.1.1	Required	n/a	n/a	IDB	Yes	Yes	Required	
D.1.2	Advisory	n/a	n/a	IDB	Yes	Yes	Required	intended to apply to experimental and unstable features, forcing full documentation
D.2.1	Required	n/a	n/a	UB, CQ, DC	Yes	Yes	Required	
D.3.1	Required	n/a	n/a	CQ	Yes	Yes	Required	
D.4.1	Required	n/a	n/a	UB, CQ	Yes	Yes	Required	often in the form of panics
D.4.2	Advisory	n/a	n/a	IDB, CQ	Yes	No	Required	
D.4.3	Required	n/a	n/a	DC, CQ	Yes	No	Advisory	
D.4.4	Advisory	n/a	n/a	DC	Yes	Yes	Advisory	conditional compilation is provided by the cfg attribute
D.4.5	Advisory	n/a	n/a	DC	Yes	Yes	Advisory	“ambiguity” is determined by the project
D.4.6	Advisory	n/a	n/a	DC	No	No	n/a	all primitive types already fulfil this
D.4.7	Required	n/a	n/a	DC	Yes	Yes	Advisory	prefer Option, Result, etc. over in-band error values
D.4.8	Advisory	n/a	n/a	DC	No	No	n/a	
D.4.9	Advisory	n/a	n/a	DC, CQ	Yes	Yes	Advisory	
D.4.10	Required	n/a	n/a	UB, DC	No	No	n/a	
D.4.11	Required	n/a	n/a	UB, IDB	Yes	Yes	Advisory	
D.4.12	Required	n/a	n/a	UB, CQ	Yes	Yes	Advisory	
D.4.13	Advisory	n/a	n/a	UB, DC	Yes	Yes	Required	many Rust APIs use the type system to enforce ordering
D.4.14	Required	n/a	n/a	UB, CQ	Yes	Yes	Required	
D.4.15	Required	n/a	n/a	UB, IDB, DC	Yes	Yes	Required	Rust implements IEEE-754
D.5.1	Required	n/a	n/a	UB	Yes	Partial	Required	not all safe Rust types are race-free
D.5.2	Required	n/a	n/a	UB	Yes	Yes	Required	
D.5.3	Required	n/a	n/a	UB, DC	Yes	Yes	Required	

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
Rules								
R.1.1	Required	Decidable	STU	UB, IDB	Yes	Partial	Required	
R.1.3	Required	Undecidable	System	UB, IDB	Yes	Yes	Required	
R.1.4	Required	Decidable	STU	UB, DC	No	No	n/a	this is specific to C versioning
R.1.5	Required	Undecidable	System	UB, IDB, DC	Yes	Yes	Required	this applies to deprecated APIs
R.2.1	Required	Undecidable	System	DC	Yes	Yes	Advisory	
R.2.2	Required	Undecidable	System	DC	Yes	Yes	Required	
R.2.3	Advisory	Decidable	System	DC	Yes	Yes	Advisory	
R.2.4	Advisory	Decidable	System	DC	No	No	n/a	no separate tag name space in Rust
R.2.5	Advisory	Decidable	System	DC	Yes	Yes	Advisory	
R.2.6	Advisory	Decidable	STU	DC	Yes	Yes	Advisory	
R.2.7	Advisory	Decidable	STU	DC	Yes	Yes	Advisory	
R.2.8	Advisory	Decidable	System	DC	Yes	Yes	Advisory	
R.3.1	Required	Decidable	STU	DC	Yes	Yes	Advisory	nested comments are fully supported
R.3.2	Required	Decidable	STU	DC	No	No	n/a	
R.4.1	Required	Decidable	STU	DC, IDB	No	No	n/a	
R.4.2	Advisory	Decidable	STU	DC	No	No	n/a	
R.5.1	Required	Decidable	System	UB, IDB, DC	Yes	Partial	Required	no character limit, except in extern “C”, but one can be set by project
R.5.2	Required	Decidable	STU	UB, IDB, CQ	Yes	Yes	Required	no character limit, but one can be applied; has name spaces
R.5.3	Required	Decidable	STU	DC	Yes	Yes	Advisory	this also applies to macro names
R.5.4	Required	Decidable	STU	UB, IDB, DC	No	No	n/a	
R.5.5	Required	Decidable	STU	UB, IDB, DC	Partial	Partial	Advisory	macros and functions use different syntax
R.5.6	Required	Decidable	System	DC	Yes	Yes	Advisory	the proper module system makes surprise name conflicts much less likely
R.5.7	Required	Decidable	System	UB, DC	No	No	n/a	no separate tag name space in Rust
R.5.8	Required	Decidable	System	DC	Yes	Yes	Advisory	the proper module system makes surprise name conflicts much less likely
R.5.9	Advisory	Decidable	System	DC	Yes	Yes	Advisory	

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
R.5.10	Required	Decidable	STU	UB, DC	Yes	Partial	Advisory	only possible in some cases. Previously Rule 21.2
R.6.1	Required	Decidable	STU	UB, IDB	No	No	n/a	only provided as a library feature
R.6.2	Required	Decidable	STU	DC	No	No	n/a	
R.6.3	Required	Decidable	STU	IDB	No	No	n/a	
R.7.1	Required	Decidable	STU	DC	Yes	Yes	Advisory	Rust octals have a distinct prefix from decimals
R.7.2	Required	Decidable	STU	DC	Yes	Yes	Advisory	this is an error by default but can be enabled. Note that suffixes also make the size explicit
R.7.3	Required	Decidable	STU	DC	No	No	n/a	
R.7.4	Required	Decidable	STU	UB	No	No	n/a	
R.7.5	Mandatory	Decidable	STU	UB	No	No	n/a	
R.7.6	Required	Decidable	STU	DC	No	No	n/a	
R.8.1	Required	Decidable	STU	DC	No	No	n/a	
R.8.2	Required	Decidable	STU	UB, DC	No	No	n/a	
R.8.3	Required	Decidable	System	UB, DC	Yes	No	Required	an extern declaration shall have a type compatible with the C declaration
R.8.4	Required	Decidable	STU	UB	No	No	n/a	
R.8.5	Required	Decidable	System	DC	Yes	No	Advisory	may affect extern “C” declarations
R.8.6	Required	Decidable	System	UB	Yes	No	Required	may affect extern “C” declarations
R.8.7	Advisory	Decidable	System	DC	Yes	Yes	Advisory	items should not be declared pub if referenced in only one module
R.8.8	Required	Decidable	STU	DC	No	No	n/a	
R.8.9	Advisory	Decidable	System	DC	Yes	Yes	Advisory	
R.8.10	Required	Decidable	STU	UB, DC	No	No	n/a	
R.8.11	Advisory	Decidable	STU	DC	No	No	n/a	
R.8.12	Required	Decidable	STU	DC	No	No	n/a	
R.8.13	Advisory	Undecidable	System	DC	Yes	Yes	Advisory	mut should be avoided unless necessary
R.8.14	Required	Decidable	STU	UB	No	No	n/a	
R.8.15	Required	Decidable	System	UB	Yes	No	Required	may affect extern “C” declarations

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
R.8.16	Advisory	Decidable	STU	DC	No	No	n/a	cannot be explicitly specified. Only ZSTs have this alignment
R.8.17	Advisory	Decidable	STU	DC	Partial	Partial	Advisory	alignment applies to types, not objects
R.8.18	Required	Decidable	STU	UB, DC	No	No	n/a	
R.8.19	Advisory	Decidable	STU	UB, DC	No	No	n/a	
R.9.1	Mandatory	Undecidable	System	UB	Yes	Yes	Required	enforced by rustc but can be bypassed by unsafe
R.9.2	Required	Decidable	STU	UB, CQ, DC	No	No	n/a	
R.9.3	Required	Decidable	STU	UB	No	No	n/a	
R.9.4	Required	Decidable	STU	DC	Yes	Yes	Required	enforced by rustc
R.9.5	Required	Decidable	STU	IDB, DC	No	No	n/a	
R.9.6	Required	Decidable	STU	DC	No	No	n/a	
R.9.7	Mandatory	Undecidable	System	UB	Yes	No	Required	
R.10.1	Required	Decidable	STU	UB, IDB, DC	No	No	n/a	
R.10.2	Required	Decidable	STU	DC	No	No	n/a	
R.10.3	Required	Decidable	STU	UB, IDB	No	No	n/a	
R.10.4	Required	Decidable	STU	IDB	No	No	n/a	
R.10.5	Advisory	Decidable	STU	DC	Yes	Partial	Advisory	includes both safe `as` and unsafe `transmute` operations
R.10.6	Required	Decidable	STU	DC	No	No	n/a	
R.10.7	Required	Decidable	STU	DC	No	No	n/a	
R.10.8	Required	Decidable	STU	DC	Yes	Partial	Advisory	includes both safe `as` and unsafe `transmute` operations
R.11.1	Required	Decidable	STU	UB, IDB	Yes	Partial	Required	includes both safe `as` and unsafe `transmute` operations
R.11.2	Required	Decidable	STU	UB	Yes	No	Required	
R.11.3	Required	Decidable	STU	UB	Yes	Yes	Required	
R.11.4	Advisory	Decidable	STU	UB, IDB	Yes	Yes	Required	
R.11.5	Advisory	Decidable	STU	UB	Yes	No	Required	
R.11.6	Required	Decidable	STU	UB, IDB	Yes	No	Advisory	
R.11.8	Required	Decidable	STU	UB	Yes	Partial	Required	

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
R.11.9	Required	Decidable	STU	DC	No	No	n/a	Rust does not have a null pointer constant (specific concept to C)
R.11.10	Required	Decidable	STU	UB	No	No	n/a	
R.11.11	Advisory	Decidable	STU	DC	Yes	Yes	Required	enforced by rustc
R.12.1	Advisory	Decidable	STU	DC	Yes	Yes	Advisory	
R.12.2	Required	Undecidable	System	UB, DC	Yes	Partial	Advisory	
R.12.3	Advisory	Decidable	STU	DC	No	No	n/a	
R.12.4	Advisory	Decidable	STU	DC	Yes	No	Advisory	this is either well-defined or will not occur
R.12.5	Mandatory	Decidable	STU	DC	No	No	n/a	
R.12.6	Required	Decidable	STU	UB	No	No	n/a	
R.13.1	Required	Undecidable	System	UB	Yes	Yes	Advisory	order of evaluation is strict in Rust
R.13.2	Required	Undecidable	System	UB	No	No	n/a	order of evaluation is strict in Rust
R.13.3	Advisory	Decidable	STU	UB, DC	No	No	n/a	
R.13.4	Advisory	Decidable	STU	UB, DC	No	No	n/a	result has unit type and order of evaluation is strict in Rust
R.13.5	Required	Undecidable	System	DC	Yes	Yes	Advisory	
R.13.6	Required	Decidable	STU	UB, DC	No	No	n/a	this is not an expression operator in Rust
R.14.1	Required	Undecidable	System	DC	Yes	Partial	Required	applies to while loops only
R.14.2	Required	Undecidable	System	DC	No	No	n/a	
R.14.3	Required	Undecidable	System	DC	Yes	Yes	Required	
R.14.4	Required	Decidable	STU	DC	Yes	Yes	Required	enforced by rustc
R.15.1	Advisory	Decidable	STU	DC	No	No	n/a	
R.15.2	Required	Decidable	STU	DC	No	No	n/a	
R.15.3	Required	Decidable	STU	DC	No	No	n/a	
R.15.4	Advisory	Decidable	STU	DC	Yes	Yes	Advisory	
R.15.5	Advisory	Decidable	STU	DC	Yes	Yes	Disapplied	
R.15.6	Required	Decidable	STU	DC	No	No	n/a	
R.15.7	Required	Decidable	STU	DC	Yes	Yes	Advisory	

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
R.16.1	Required	Decidable	STU	DC	No	No	n/a	
R.16.2	Required	Decidable	STU	DC	No	No	n/a	
R.16.3	Required	Decidable	STU	DC	No	No	n/a	
R.16.4	Required	Decidable	STU	DC	No	No	n/a	a corresponding match expression must be complete
R.16.5	Required	Decidable	STU	DC	No	No	n/a	irrefutable pattern causes a subsequent refutable one to be unreachable
R.16.6	Required	Decidable	STU	DC	No	No	n/a	
R.16.7	Required	Decidable	STU	DC	No	No	n/a	
R.17.1	Required	Decidable	STU	UB	No	No	n/a	
R.17.2	Required	Undecidable	System	UB, DC	Yes	Yes	Required	
R.17.3	Mandatory	Decidable	STU	UB	No	No	n/a	
R.17.4	Mandatory	Decidable	STU	UB	No	No	n/a	the return keyword is not needed to return a value in Rust, only to exit
R.17.5	Required	Undecidable	System	UB, DC	No	No	n/a	
R.17.7	Required	Decidable	STU	DC	Yes	Yes	Required	must_use can help indicate where this is important, but does not affect applicability.
R.17.8	Advisory	Undecidable	System	DC	Yes	Yes	Disapplied	this cannot be done accidentally without declaring parameters `mut`
R.17.9	Mandatory	Undecidable	System	UB	Yes	No	Required	this is expressed with the `!` (Never) type, and enforced by rustc
R.17.10	Required	Decidable	STU	DC	No	No	n/a	
R.17.11	Advisory	Undecidable	System	DC	Yes	Yes	Advisory	a non-returning function can be declared to return a value type
R.17.12	Advisory	Decidable	STU	DC	No	No	n/a	
R.17.13	Required	Decidable	STU	UB	No	No	n/a	
R.18.1	Required	Undecidable	System	UB	Yes	No	Required	by unsafe API
R.18.2	Required	Undecidable	System	UB	Yes	No	Required	by unsafe API
R.18.3	Required	Undecidable	System	UB	Yes	Yes	Required	
R.18.4	Advisory	Decidable	STU	DC	Yes	No	Advisory	applies to use of the unsafe API
R.18.5	Advisory	Decidable	STU	DC	Yes	Yes	Advisory	
R.18.6	Required	Undecidable	System	UB	Yes	No	Required	
R.18.7	Required	Decidable	STU	UB, DC	No	No	n/a	

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
R.18.8	Required	Decidable	STU	UB, DC	No	No	n/a	
R.18.9	Required	Decidable	STU	UB	No	No	n/a	
R.18.10	Mandatory	Decidable	STU	UB	No	No	n/a	
R.19.1	Mandatory	Undecidable	System	UB	Yes	No	Required	
R.19.2	Advisory	Decidable	STU	UB, DC	Yes	Yes	Advisory	
R.19.3	Required	Undecidable	System	UB	Yes	Yes	Required	
R.20.1	Advisory	Decidable	STU	UB	No	No	n/a	rules specific to the C preprocessor do not apply to Rust
R.20.2	Required	Decidable	STU	UB	No	No	n/a	
R.20.3	Required	Decidable	STU	UB	No	No	n/a	
R.20.4	Required	Decidable	STU	UB	Partial	Partial	Required	possible with raw identifiers but the compiler prevents visual conflicts
R.20.5	Advisory	Decidable	STU	DC	No	No	n/a	
R.20.6	Required	Decidable	STU	UB	No	No	n/a	
R.20.7	Required	Decidable	STU	DC	Partial	Partial	Advisory	possible to express with procedural macros only, not macro_rules
R.20.8	Required	Decidable	STU	DC	No	No	n/a	
R.20.9	Required	Decidable	STU	DC	No	No	n/a	
R.20.10	Advisory	Decidable	STU	UB	No	No	n/a	
R.20.11	Required	Decidable	STU	UB	No	No	n/a	
R.20.12	Required	Decidable	STU	DC	No	No	n/a	
R.20.13	Required	Decidable	STU	DC	No	No	n/a	
R.20.14	Required	Decidable	STU	DC	No	No	n/a	
R.20.15	Required	Decidable	STU	UB	No	No	n/a	
R.21.3	Required	Decidable	STU	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.4	Required	Decidable	STU	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.5	Required	Decidable	STU	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.6	Required	Decidable	STU	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.7	Required	Decidable	STU	UB	Yes	No	Required	only accessible through unsafe extern “C”

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
R.21.8	Required	Decidable	STU	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.9	Required	Decidable	STU	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.10	Required	Decidable	STU	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.11	Advisory	Decidable	STU	UB	No	No	n/a	no external interface
R.21.12	Required	Decidable	STU	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.13	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.14	Required	Undecidable	System	DC	Yes	No	Required	only accessible through unsafe extern “C”
R.21.15	Required	Decidable	STU	DC	Yes	No	Required	only accessible through unsafe extern “C”
R.21.16	Required	Decidable	STU	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.17	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.18	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.19	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.20	Mandatory	Undecidable	System	IDB, DC	Yes	No	Required	only accessible through unsafe extern “C”
R.21.21	Required	Decidable	STU	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.21.22	Mandatory	Decidable	STU	UB	No	No	n/a	no external interface
R.21.23	Required	Decidable	STU	DC	No	No	n/a	no external interface
R.21.24	Required	Decidable	STU	CQ	Yes	No	Required	only accessible through unsafe extern “C”
R.21.25	Required	Decidable	STU	UB	Yes	Yes	Required	
R.21.26	Required	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.1	Required	Undecidable	System	UB, CQ	Yes	No	Required	applies to resources acquired through FFI only
R.22.2	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.3	Required	Undecidable	System	UB, IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.4	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.5	Mandatory	Undecidable	System	IDB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.6	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.7	Required	Undecidable	System	DC	Yes	No	Required	only accessible through unsafe extern “C”

Guideline	MISRA C Status				Rust Applicability		Rust Assessment	
	Category	Decidability	Scope	Rationale	(1)	(2)	Adjusted Category	Comment
R.22.8	Required	Undecidable	System	DC	Yes	No	Disapplied	only accessible through unsafe extern “C”
R.22.9	Required	Undecidable	System	DC	Yes	No	Disapplied	only accessible through unsafe extern “C”
R.22.10	Required	Undecidable	System	DC	Yes	No	Disapplied	only accessible through unsafe extern “C”
R.22.11	Required	Undecidable	System	UB	Yes	Partial	Required	
R.22.12	Mandatory	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.13	Required	Decidable	STU	UB, DC	Yes	Yes	Required	
R.22.14	Mandatory	Undecidable	System	UB	Yes	Partial	Required	applies to creating synchronization objects before threads that use them
R.22.15	Required	Undecidable	System	UB	Yes	Partial	Required	applies to releasing synchronization objects after threads that use them
R.22.16	Required	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.17	Required	Undecidable	System	UB	Yes	No	Required	only accessible through unsafe extern “C”
R.22.18	Required	Undecidable	System	UB	Yes	Yes	Required	
R.22.19	Required	Undecidable	System	UB	Yes	Yes	Required	
R.22.20	Mandatory	Undecidable	System	UB	Yes	Partial	Required	
R.23.1	Advisory	Decidable	STU	DC	No	No	n/a	
R.23.2	Required	Decidable	STU	DC	No	No	n/a	
R.23.3	Advisory	Decidable	STU	DC	No	No	n/a	
R.23.4	Required	Decidable	STU	DC	No	No	n/a	
R.23.5	Advisory	Decidable	STU	DC	No	No	n/a	
R.23.6	Required	Decidable	STU	DC	No	No	n/a	
R.23.7	Advisory	Decidable	STU	DC	No	No	n/a	
R.23.8	Required	Decidable	STU	DC	No	No	n/a	

4 References

[1] MISRA C:2025 *Guidelines for the use of the C language in critical systems*
ISBN 978-1-911700-19-7 (paperback), ISBN 978-1-911700-20-3 (PDF)
The MISRA Consortium Limited, Norwich, March 2025

[2] The Rust Programming Language, 2nd Edition, Klabnik, S. and Nichols, C.,
ISBN 978-1-718503-10-6
No Starch Press, 2022 (with updates for Rust 1.81.0, 2024)