

ISO/IEC JTC 1/SC 22/OWG V N0105

Examples of Mapping MISRA-C Rules to COBOL

Date: 26 October 2007

Contributed by: Barry Tauber

Original file name: OWG20Rules.doc

Notes:

Hi Jim-

I don't know if useful, but I have annotated the attached table with some COBOL examples.

Thanks,

Barry

The 20 rules identified so far from MISRA-C to reduce vulnerabilities.

Abbreviation is the OWG tag for template.

Potential for Language Independence subjective.

COBOL Pseudo simple example of how rule would apply in COBOL.

Barry

Number	Abbreviation	Existing Rule	Potential for Language Independence	Use as is or modified rule
1	S_GLOB	There shall be no dependence on any of the undefined features of ISO C	Yes	There shall be no dependence on any of the undefined features of the language standard.
COBOL Pseudo		No MOVE ABC TO ABC.		
2	F_PROT	All function calls and definitions shall be preceded by a new-style prototype	Yes, in a broadened sense	Deprecated features of the language should not be used.
COBOL Pseudo		Avoid "NEXT SENTENCE" , USE CONTINUE or EXIT PERFORM		
3	F_COMP	All function call arguments and function definition parameters shall be compatible with the corresponding arguments in the corresponding function declaration	Yes	As is
COBOL Pseudo		Picture Clauses in USING should match in Caller and Callee		

Number	Abbreviation	Existing Rule	Potential for Language Independence	Use as is or modified rule
4	E_PREC	A boolean-valued sub-expression shall not appear next to a bit operator, an assignment operator or a relational operator in an if or while expression	Yes	As is
COBOL Pseudo		No IF (A OR B = "C"), USE IF ((A) OR (B = "C"))		
5	E_SIDE	Every expression statement shall have at least one side-effect for any execution path	Yes	As is
COBOL Pseudo		No Compute A = A ; Rule says something should change		
6	E_COMM	The expression on the left hand side of a comma operator shall have at least one side-effect	No, too language specific	
COBOL Pseudo		No example, too specific to C language.		
7	E_UNEG	An unsigned expression shall not be compared for negativity	Yes, include in an integer coercion rule	Integer coercion, such as comparing an unsigned expression for negativity, shall not be done.
COBOL Pseudo		If A PIC 9(4), then no IF A < ZERO statements		
8	E_CSIGN	No implicit conversion shall change the signed nature of an object or reduce the number of bits in that object	Yes	As is
COBOL Pseudo		a) A PIC X(20), then B REDEFINES A, B should have full 20 characters, use FILLER where necessary b) no E REDEFINES F, where E PIC 9(5), F PIC S9(5) c) no MOVE C to D, where C PIC S9(5), D PIC 9(3).		
9	E_REACH	Every non-null statement shall be reachable	Yes	All statements should be reachable
COBOL Pseudo		Obvious.		
10	E_HIDE	Local objects shall not hide objects with internal or external linkage	Yes	
COBOL Pseudo		COPY statement contents should not include CALL statements		
11	E_EXTR	The extern keyword shall not be used in a nested block	Yes	

Number	Abbreviation	Existing Rule	Potential for Language Independence	Use as is or modified rule
	COBOL Pseudo	EXTERNAL should only be in 01 level And EXTERNAL item should not be included in a USING clause		
12	E_FEQL	Floating point objects shall not be compared for equality or inequality	Yes	As is
	COBOL Pseudo	Obvious		
13	E_FLOOP	Floating point objects shall not be used in the initialization, controlling or re-initialization expressions of a for statement or the controlling expression of a while statement.	Yes	As is
	COBOL Pseudo	Floating point not to be used as subscript or VARYING item		
14	E_BITF1	Every named bitfield of size 1 shall use the unsigned or signed keyword in its declaration	Yes	As is
	COBOL Pseudo	More C language specific.		
15	E_NARROW	No pointer shall be cast to a narrower integral type	Yes	As is
	COBOL Pseudo	a) Do not even try to REDEFINE a POINTER item b) Be cautious using "AS" clause		
16	E_PCAST	Pointers shall not be cast to different pointer types	Yes	As is
	COBOL Pseudo	Do not even try to REDEFINE a POINTER item		
17	E_CASE1	A switch statement shall have at least one 'case:' and shall have exactly one 'default:'	Yes	As is
	COBOL Pseudo	EVALUATE should have at least one condition and should contain the OTHER clause		
18	E_CFALL	No 'case:' or 'default:' shall be reachable from the previous 'case:' or 'default:' except by direct fall-through	Yes	As is
	COBOL Pseudo	Avoid GO TO within EVALUATE		

Number	Abbreviation	Existing Rule	Potential for Language Independence	Use as is or modified rule
19	P_ARGS2	No function-like macro shall use an argument more than once	Yes	
COBOL Pseudo		COPY with REPLACING should contain unique values		
20	P_PAREN	All macro arguments shall be parenthesized unless by doing so a syntax violation would be created	No	
COBOL Pseudo		Mostly C language issue		