

“Power of Ten” Coverage for CodeSonar®

The “Power of Ten” refers to a set of ten rules developed by Gerard Holzmann of the NASA Jet Propulsion Laboratory for use in writing safety-critical software. The rules are simple, but they specify strict limits on the forms code can take. These limits support code clarity and analyzability, which are especially important for safety-critical applications.

“Power of Ten” Rules Detected by CodeSonar:

The following table lists the CodeSonar warning classes that are associated with “Power of Ten” rules. The tenth rule, POW10:10, is associated with some checks and also requires the use of a static analysis tool such as CodeSonar.

“Power of Ten” Rule	CodeSonar Mnemonic	CodeSonar Warning Class
<u>1</u>	LANG.STRUCT.GOTO	Goto Statement
	LANG.FUNCS.RECURSION	Recursion
	BADFUNC.LONGJMP	Use of longjmp
	BADFUNC.SETJMP	Use of setjmp
<u>2</u>	LANG.STRUCT.ULOOP	Potential Unbounded Loop
<u>3</u>	ALLOC.POSTINIT	Dynamic Allocation After Initialization
<u>4</u>	LANG.FUNCS.TOOLONG	Function Too Long
<u>5</u>	LANG.FUNCS.ASSERTS	Not Enough Assertions
<u>6</u>	LANG.STRUCT.SCOPE.FILE	Scope Could Be File Static
	LANG.STRUCT.SCOPE.LOCAL	Scope Could Be Local Static
<u>7</u>	LANG.FUNCS.IRV	Ignored Return Value
	LANG.STRUCT.UPD	Unchecked Parameter Dereference
<u>8</u>	LANG.PREPROC.COND	Conditional Compilation
	LANG.PREPROC.MACROEND	Macro Does Not End With } or)
	LANG.PREPROC.MACROSTART	Macro Does Not Start With { or (
	LANG.PREPROC.PASTE	Macro Uses ## Operator
	LANG.PREPROC.RECURSIVE	Recursive Macro
	LANG.PREPROC.UNBALANCED	Unbalanced Parenthesis
	LANG.PREPROC.VARIADIC	Variadic Macro
<u>9</u>	LANG.STRUCT.FUNCPTR.CALL	Function Pointer
	LANG.PREPROC.ARROW	Macro Uses -> Operator
	LANG.PREPROC.BRACES	Macro uses [] Operator
	LANG.PREPROC.STAR	Macro Uses Unary * Operator
	LANG.STRUCT.PIT	Pointer Type Inside Typedef
	LANG.STRUCT.TMD	Too Many Dereferences
<u>10</u>	BUILD.WALL	Not All Warnings Are Enabled
	BUILD.WERROR	Warnings Not Treated As Errors