

CQA Supported Rules

The following rules can be enabled in CQA mode:

- CS.BRM.AIBA - Avoid 'is' before 'as'
- CS.BRM.BEB - Avoid block statements with empty bodies
- CS.BRM.CCB - Always enclose if and else bodies with curly braces
- CS.BRM.CPEB - Avoid checked, unchecked, fixed and unsafe statements with empty bodies
- CS.BRM.ETK - Use keyword 'this' explicitly when accessing capitalized members
- CS.BRM.ICB - Always enclose iteration statements bodies with curly braces
- CS.BRM.KFATNC - Keep file and type names consistent
- CS.BRM.SWDEFLAST - Place 'default' as the last case of the 'switch' statement
- CS.CDD.DUPU - Avoid duplicate using statements
- CS.EU.VZS - Ensure that each enum has member with value 0
- CS.INTER.ITT - String literals should be internationalized
- CS.NG.VAR.PNCFV - Follow proper naming convention for method variables and consts.
- CS.PB.USC.CC - Avoid unreachable code in condition
- CS.PB.USC.UC - Avoid unreachable code
- CS.PB.ANIL - Avoid non-iterable loops
- CS.PB.AWC - Avoid assignment within a condition
- CS.PB.BITBOOL - Do not use bitwise operators on bool operands
- CS.PB.CCA - Avoid confusing assignments to constructor arguments
- CS.PB.CEB - Avoid conditional statements with empty bodies
- CS.PB.CNFA - Check for 'null' when using 'as' operator
- CS.PB.EEB - Avoid try, catch, finally and using statements with empty bodies
- CS.PB.FPLI - Do not use floating point variables as loop indices.
- CS.PB.IDNE - Avoid increment and decrement statements which have no effect
- CS.PB.IEB - Avoid iteration statements with empty bodies
- CS.PB.IVFLC - Use initializer variable in a condition of 'for' loop
- CS.PB.IVFLI - Use initializer variable in 'for' loop iterator section
- CS.PB.MCO - Review '?' operator for potential misuse
- CS.PB.NACC - Avoid inaccessible classes and structs
- CS.PB.NSIVFLB - Do not modify 'for' loop initializer variable in 'for' loop body
- CS.PB.NSIVFLN - Do not increment or decrement on the same variable over multiple nested 'for' loop statements
- CS.PB.PUO - Avoid using the unary + operator
- CS.PB.VTNV - Do not compare value types to null
- CS.PB.WIBS - Avoid wrong indentation of blockless statements
- CS.SEC.RSFSS - Do not re-create string from SecureString.
- CS.SEC.SSFP - Use SecureString instead of String for password.
- CS.TRS.LCB - Always enclose lock statement body with curly braces
- CS.TRS.LEB - Avoid lock statements with empty bodies.
- CWE.316.RSFSS - Do not re-create string from SecureString
- CWE.316.SSFP - Use SecureString instead of String for password
- CWE.476.CNFA - Check for 'null' when using 'as' operator
- CWE.480.PUO - Avoid using the unary + operator
- CWE.481.AWC - Avoid assignment within a condition
- IFD.IDDR - Implement IDisposable in types which are using disposable resources
- OWASP2017.A3.RSFSS - Do not re-create string from SecureString
- OWASP2017.A3.SSFP - Use SecureString instead of String for password

