

Id	Title	Severity	Description
Nullable types			
S1696	NullReferenceException should not be caught	warning	https://rules.sonarsource.com/csharp/RSPEC-1696
S2259	Null pointers should not be dereferenced	warning	https://rules.sonarsource.com/csharp/RSPEC-2259
S3610	Nullable type comparison should not be redundant	warning	https://rules.sonarsource.com/csharp/RSPEC-3610
S3655	Empty nullable value should not be accessed	warning	https://rules.sonarsource.com/csharp/RSPEC-3655
Conventions			
CA1024	Use properties where appropriate	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1024
CA1036	Override methods on comparable types	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1036
CA1050	Declare types in namespaces	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1050
CA2200	Rethrow to preserve stack details	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca2200
CA2226	Operators should have symmetrical overloads	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca2226
RCS1210	Return completed task instead of returning null.	warning	http://pihrt.net/roslynator/analyzer?id=RCS1210
S1048	Destructors should not throw exceptions	warning	https://rules.sonarsource.com/csharp/RSPEC-1048
S1104	Fields should not have public accessibility	warning	https://rules.sonarsource.com/csharp/RSPEC-1104
S1163	Exceptions should not be thrown in finally blocks	warning	https://rules.sonarsource.com/csharp/RSPEC-1163
S1206	"Equals(Object)" and "GetHashCode()" should be overridden in pairs	warning	https://rules.sonarsource.com/csharp/RSPEC-1206
S2328	"GetHashCode" should not reference mutable fields	warning	https://rules.sonarsource.com/csharp/RSPEC-2328
S2345	Flags enumerations should explicitly initialize all their members	warning	https://rules.sonarsource.com/csharp/RSPEC-2345
S3400	Methods should not return constants	warning	https://rules.sonarsource.com/csharp/RSPEC-3400
S3453	Classes should not have only "private" constructors	warning	https://rules.sonarsource.com/csharp/RSPEC-3453
S3887	Mutable, non-private fields should not be "readonly"	warning	https://rules.sonarsource.com/csharp/RSPEC-3887
S3903	Types should be defined in named namespaces	warning	https://rules.sonarsource.com/csharp/RSPEC-3903
S3984	Exceptions should not be created without being thrown	warning	https://rules.sonarsource.com/csharp/RSPEC-3984
CA1507	Use nameof to express symbol names	suggestion	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1507
CA1710	Identifiers should have correct suffix	suggestion	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1710
S100	Methods and properties should be named in PascalCase	suggestion	https://rules.sonarsource.com/csharp/RSPEC-100
S101	Types should be named in PascalCase	suggestion	https://rules.sonarsource.com/csharp/RSPEC-101
S3256	"string.IsNullOrEmpty" should be used	suggestion	https://rules.sonarsource.com/csharp/RSPEC-3256

Id	Title	Severity	Description
Incorrect assignment to variable or field			
S1226	Method parameters, caught exceptions and foreach variables' initial values should not be ignored	warning	https://rules.sonarsource.com/csharp/RSPEC-1226
S1656	Variables should not be self-assigned	warning	https://rules.sonarsource.com/csharp/RSPEC-1656
S1764	Identical expressions should not be used on both sides of a binary operator	warning	https://rules.sonarsource.com/csharp/RSPEC-1764
S2123	Values should not be uselessly incremented	warning	https://rules.sonarsource.com/csharp/RSPEC-2123
S2183	Integral numbers should not be shifted by zero or more than their number of bits-1	warning	https://rules.sonarsource.com/csharp/RSPEC-2183
S2184	Results of integer division should not be assigned to floating point variables	warning	https://rules.sonarsource.com/csharp/RSPEC-2184
S2201	Return values from functions without side effects should not be ignored	warning	https://rules.sonarsource.com/csharp/RSPEC-2201
S2225	"ToString()" method should not return null	warning	https://rules.sonarsource.com/csharp/RSPEC-2225
S2234	Parameters should be passed in the correct order	warning	https://rules.sonarsource.com/csharp/RSPEC-2234
S2757	"+=" should not be used instead of "+="	warning	https://rules.sonarsource.com/csharp/RSPEC-2757
S2761	Doubled prefix operators "!!" and "~~" should not be used	warning	https://rules.sonarsource.com/csharp/RSPEC-2761
S2995	"Object.ReferenceEquals" should not be used for value types	warning	https://rules.sonarsource.com/csharp/RSPEC-2995
S3169	Multiple "OrderBy" calls should not be used	warning	https://rules.sonarsource.com/csharp/RSPEC-3169
S3949	Calculations should not overflow	warning	https://rules.sonarsource.com/csharp/RSPEC-3949
S4275	Getters and setters should access the expected fields	warning	https://rules.sonarsource.com/csharp/RSPEC-4275
Performance and resource management			
CA1001	Types that own disposable fields should be disposable	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1001
RCS1242	Do not pass non-read-only struct by read-only reference.	warning	http://pihrt.net/roslynator/analyzer?id=RCS1242
S1643	Strings should not be concatenated using '+' in a loop	warning	https://rules.sonarsource.com/csharp/RSPEC-1643
S2930	"IDisposable" should be disposed	warning	https://rules.sonarsource.com/csharp/RSPEC-2930
S2931	Classes with "IDisposable" members should implement "IDisposable"	warning	https://rules.sonarsource.com/csharp/RSPEC-2931
S2952	Classes should "Dispose" of members from the classes' own "Dispose" methods	warning	https://rules.sonarsource.com/csharp/RSPEC-2952
S2997	"IDisposable" created in a "using" statement should not be returned	warning	https://rules.sonarsource.com/csharp/RSPEC-2997
Conditional executions, iterations and recursions			
CA1508	Avoid dead conditional code	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1508
CA2245	Do not assign a property to itself	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca2245
RCS1215	Expression is always equal to true/false.	warning	http://pihrt.net/roslynator/analyzer?id=RCS1215
S1244	Floating point numbers should not be tested for equality	warning	https://rules.sonarsource.com/csharp/RSPEC-1244
S1751	Loops with at most one iteration should be refactored	warning	https://rules.sonarsource.com/csharp/RSPEC-1751
S1862	Related "if/else if" statements should not have the same condition	warning	https://rules.sonarsource.com/csharp/RSPEC-1862

Id	Title	Severity	Description
S1871	Two branches in a conditional structure should not have exactly the same implementation	warning	https://rules.sonarsource.com/csharp/RSPEC-1871
S2190	Recursion should not be infinite	warning	https://rules.sonarsource.com/csharp/RSPEC-2190
S2251	A "for" loop update clause should move the counter in the right direction	warning	https://rules.sonarsource.com/csharp/RSPEC-2251
S2252	For-loop conditions should be true at least once	warning	https://rules.sonarsource.com/csharp/RSPEC-2252
S2583	Conditionally executed code should be reachable	warning	https://rules.sonarsource.com/csharp/RSPEC-2583
S2589	Boolean expressions should not be gratuitous	warning	https://rules.sonarsource.com/csharp/RSPEC-2589
S2688	"NaN" should not be used in comparisons	warning	https://rules.sonarsource.com/csharp/RSPEC-2688
S3923	All branches in a conditional structure should not have exactly the same implementation	warning	https://rules.sonarsource.com/csharp/RSPEC-3923
S3981	Collection sizes and array length comparisons should make sense	warning	https://rules.sonarsource.com/csharp/RSPEC-3981
S4158	Empty collections should not be accessed or iterated	warning	https://rules.sonarsource.com/csharp/RSPEC-4158
Polymorphism and inheritance			
CA1061	Do not hide base class methods	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1061
CA2214	Do not call overridable methods in constructors	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca2214
S3249	Classes directly extending "object" should not call "base" in "GetHashCode" or "Equals"	warning	https://rules.sonarsource.com/csharp/RSPEC-3249
S3397	base.Equals" should not be used to check for reference equality in "Equals" if "base" is not "object"	warning	https://rules.sonarsource.com/csharp/RSPEC-3397
S3464	Type inheritance should not be recursive	warning	https://rules.sonarsource.com/csharp/RSPEC-3464
S3466	Optional parameters should be passed to "base" calls	warning	https://rules.sonarsource.com/csharp/RSPEC-3466
Asynchronous programming			
S3168	async" methods should not return "void"	warning	https://rules.sonarsource.com/csharp/RSPEC-3168
S4586	Non-async "Task/Task<T>" methods should not return null	warning	https://rules.sonarsource.com/csharp/RSPEC-4586
Delegates and events			
S3172	Delegates should not be subtracted	warning	https://rules.sonarsource.com/csharp/RSPEC-3172
S3244	Anonymous delegates should not be used to unsubscribe from Events	warning	https://rules.sonarsource.com/csharp/RSPEC-3244
S4583	Calls to delegate's method "BeginInvoke" should be paired with calls to "EndInvoke"	warning	https://rules.sonarsource.com/csharp/RSPEC-4583
RCS1204	Use EventArgs.Empty.	suggestion	http://pihrt.net/roslynator/analyzer?id=RCS1204
Miscellaneous			
CA1069	Enums values should not be duplicated	warning	https://docs.microsoft.com/dotnet/fundamentals/code-analysis/quality-rules/ca1069
RCS1075	Avoid empty catch clause that catches System.Exception.	warning	http://pihrt.net/roslynator/analyzer?id=RCS1075
S1848	Objects should not be created to be dropped immediately without being used	warning	https://rules.sonarsource.com/csharp/RSPEC-1848
S2114	Collections should not be passed as arguments to their own methods	warning	https://rules.sonarsource.com/csharp/RSPEC-2114

Id	Title	Severity	Description
S2275	Composite format strings should not lead to unexpected behavior at runtime	warning	https://rules.sonarsource.com/csharp/RSPEC-2275
S2674	The length returned from a stream read should be checked	warning	https://rules.sonarsource.com/csharp/RSPEC-2674
S2934	Property assignments should not be made for "readonly" fields not constrained to reference types	warning	https://rules.sonarsource.com/csharp/RSPEC-2934
S2955	Generic parameters not constrained to reference types should not be compared to "null"	warning	https://rules.sonarsource.com/csharp/RSPEC-2955
S3263	Static fields should appear in the order they must be initialized	warning	https://rules.sonarsource.com/csharp/RSPEC-3263
S3343	Caller information parameters should come at the end of the parameter list	warning	https://rules.sonarsource.com/csharp/RSPEC-3343
S3346	Expressions used in "Debug.Assert" should not produce side effects	warning	https://rules.sonarsource.com/csharp/RSPEC-3346