

epiworld

0.0-1

Generated by Doxygen 1.9.1



<b>1 Source code</b>	<b>1</b>
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>11</b>
3.1 Class List	11
<b>4 File Index</b>	<b>19</b>
4.1 File List	19
<b>5 Class Documentation</b>	<b>21</b>
5.1 Catch::Clara::accept_many_t Struct Reference	21
5.2 Action< TSeq > Struct Template Reference	21
5.2.1 Detailed Description	22
5.2.2 Constructor & Destructor Documentation	22
5.2.2.1 Action()	22
5.3 AdjList Class Reference	23
5.3.1 Constructor & Destructor Documentation	23
5.3.1.1 AdjList()	23
5.3.2 Member Function Documentation	24
5.3.2.1 read_edgelist()	24
5.4 Agent< TSeq > Class Template Reference	24
5.4.1 Detailed Description	26
5.5 Catch::Matchers::AllMatchMatcher< Matcher > Class Template Reference	27
5.6 Catch::always_false< T > Struct Template Reference	28
5.7 Catch::Matchers::AnyMatchMatcher< Matcher > Class Template Reference	29
5.8 Catch::Approx Class Reference	30
5.9 Catch::Matchers::ApproxMatcher< T, AllocComp, AllocMatch > Class Template Reference	31
5.10 Catch::Clara::Arg Class Reference	32
5.11 Catch::Clara::Args Class Reference	33
5.12 Catch::Generators::as< T > Struct Template Reference	34
5.13 Catch::AssertionHandler Class Reference	34
5.14 Catch::AssertionInfo Struct Reference	34
5.15 Catch::Detail::AssertionOrBenchmarkResult Class Reference	35
5.15.1 Detailed Description	35
5.16 Catch::AssertionReaction Struct Reference	35
5.17 Catch::AssertionResult Class Reference	36
5.18 Catch::AssertionResultData Struct Reference	37
5.19 Catch::AssertionStats Struct Reference	38
5.20 Catch::AutomakeReporter Class Reference	39
5.21 Catch::AutoReg Struct Reference	40
5.22 Catch::Clara::Detail::BasicResult< T > Class Template Reference	41
5.23 Catch::Benchmark::Benchmark Struct Reference	42

5.24 Catch::Benchmark::Detail::BenchmarkFunction Struct Reference	42
5.24.1 Detailed Description	43
5.25 Catch::BenchmarkInfo Struct Reference	43
5.26 Catch::BenchmarkStats< Duration > Struct Template Reference	44
5.27 Catch::BinaryExpr< LhsT, RhsT > Class Template Reference	45
5.28 Catch::Benchmark::Detail::bootstrap_analysis Struct Reference	46
5.29 Catch::Clara::Detail::BoundFlagLambda< L > Struct Template Reference	47
5.30 Catch::Clara::Detail::BoundFlagRef Struct Reference	49
5.31 Catch::Clara::Detail::BoundFlagRefBase Struct Reference	51
5.32 Catch::Clara::Detail::BoundLambda< L > Struct Template Reference	52
5.33 Catch::Clara::Detail::BoundManyLambda< L > Struct Template Reference	54
5.34 Catch::Clara::Detail::BoundRef Struct Reference	56
5.35 Catch::Clara::Detail::BoundValueRef< T > Struct Template Reference	57
5.36 Catch::Clara::Detail::BoundValueRef< std::vector< T > > Struct Template Reference	59
5.37 Catch::Clara::Detail::BoundValueRefBase Struct Reference	60
5.38 Catch::Captor Class Reference	61
5.39 Catch::Matchers::CasedString Struct Reference	62
5.40 Catch::Detail::CaseInsensitiveEqualTo Struct Reference	62
5.40.1 Detailed Description	62
5.41 Catch::Detail::CaseInsensitiveLess Struct Reference	62
5.41.1 Detailed Description	63
5.42 Catch_global_namespace_dummy Struct Reference	63
5.43 Catch::Benchmark::Chronometer Struct Reference	63
5.44 Catch::Benchmark::Detail::ChronometerConcept Struct Reference	63
5.45 Catch::Benchmark::Detail::ChronometerModel< Clock > Struct Template Reference	64
5.46 Catch::Generators::ChunkGenerator< T > Class Template Reference	65
5.46.1 Member Function Documentation	66
5.46.1.1 next()	66
5.47 Catch::Colour Struct Reference	67
5.48 Catch::ColourImpl::ColourGuard Class Reference	67
5.48.1 Detailed Description	67
5.48.2 Member Function Documentation	68
5.48.2.1 engage() [1/2]	68
5.48.2.2 engage() [2/2]	68
5.49 Catch::ColourImpl Class Reference	68
5.49.1 Member Function Documentation	69
5.49.1.1 guardColour()	69
5.50 Catch::TextFlow::Column Class Reference	69
5.50.1 Detailed Description	70
5.51 Catch::TextFlow::Columns Class Reference	70
5.52 Catch::CompactReporter Class Reference	71
5.52.1 Member Function Documentation	72

5.52.1.1 testRunEnded()	72
5.52.1.2 testRunStarting()	72
5.53 Catch::Benchmark::Detail::CompleteInvoker< Result > Struct Template Reference	72
5.54 Catch::Benchmark::Detail::CompleteInvoker< void > Struct Reference	73
5.55 Catch::Benchmark::Detail::CompleteType< T > Struct Template Reference	73
5.56 Catch::Benchmark::Detail::CompleteType< void > Struct Reference	73
5.57 Catch::Clara::Detail::ComposableParserImpl< DerivedT > Class Template Reference	74
5.58 Catch::Config Class Reference	75
5.59 Catch::ConfigData Struct Reference	76
5.60 Catch::Matchers::Detail::conjunction< Cond > Struct Template Reference	77
5.61 Catch::Matchers::Detail::conjunction< Cond, Rest... > Struct Template Reference	78
5.62 Catch::ConsoleReporter Class Reference	79
5.62.1 Member Function Documentation	80
5.62.1.1 testRunEnded()	81
5.62.1.2 testRunStarting()	81
5.63 Catch::TextFlow::Column::const_iterator Class Reference	81
5.63.1 Detailed Description	82
5.64 Catch::Matchers::ContainsElementMatcher< T, Equality > Class Template Reference	82
5.64.1 Detailed Description	83
5.65 Catch::Matchers::ContainsMatcher< T, AllocComp, AllocMatch > Class Template Reference	84
5.66 Catch::Matchers::ContainsMatcherMatcher< Matcher > Class Template Reference	85
5.66.1 Detailed Description	86
5.67 Catch::Counts Struct Reference	86
5.68 Catch::CumulativeReporterBase Class Reference	87
5.68.1 Detailed Description	89
5.68.2 Member Function Documentation	89
5.68.2.1 testRunEnded()	89
5.68.2.2 testRunStarting()	89
5.69 DataBase< TSeq > Class Template Reference	90
5.69.1 Detailed Description	91
5.69.2 Member Function Documentation	91
5.69.2.1 record_variant()	91
5.69.2.2 reproductive_number()	92
5.70 Catch::Decomposer Struct Reference	92
5.71 Catch::Matchers::EndsWithMatcher Class Reference	93
5.72 Entity< TSeq > Class Template Reference	94
5.73 Catch::Detail::EnumInfo Struct Reference	94
5.74 Catch::Detail::EnumValuesRegistry Class Reference	95
5.75 Catch::Benchmark::Environment< Clock > Struct Template Reference	96
5.76 Catch::Benchmark::EnvironmentEstimate< Duration > Struct Template Reference	97
5.77 Catch::Matchers::EqualsMatcher< T, AllocComp, AllocMatch > Class Template Reference	98
5.78 Catch::ErrnoGuard Class Reference	99

5.78.1 Detailed Description	99
5.79 Catch::Benchmark::Estimate< Duration > Struct Template Reference	99
5.80 Catch::EventListenerBase Class Reference	100
5.80.1 Detailed Description	101
5.80.2 Member Function Documentation	102
5.80.2.1 testRunEnded()	102
5.80.2.2 testRunStarting()	102
5.81 Catch::EventListenerFactory Class Reference	102
5.82 Catch::Matchers::ExceptionMessageMatcher Class Reference	103
5.83 Catch::ExceptionTranslatorRegistrar Class Reference	104
5.84 Catch::ExceptionTranslatorRegistry Class Reference	104
5.85 Catch::Benchmark::ExecutionPlan< Duration > Struct Template Reference	105
5.86 Catch::Clara::ExeName Class Reference	106
5.87 Catch::ExprLhs< LhsT > Class Template Reference	107
5.88 Catch::Clara::Detail::fake_arg Struct Reference	107
5.89 Catch::FatalConditionHandler Class Reference	108
5.89.1 Detailed Description	108
5.90 Catch::FatalConditionHandlerGuard Class Reference	108
5.90.1 Detailed Description	108
5.91 Catch::Generators::FilterGenerator< T, Predicate > Class Template Reference	109
5.91.1 Member Function Documentation	110
5.91.1.1 next()	110
5.92 Catch::TestSpec::FilterMatch Struct Reference	110
5.93 Catch::Generators::FixedValuesGenerator< T > Class Template Reference	111
5.93.1 Member Function Documentation	112
5.93.1.1 next()	112
5.94 Catch::GeneratorException Class Reference	112
5.95 Catch::Generators::Generators< T > Class Template Reference	113
5.95.1 Member Function Documentation	114
5.95.1.1 next()	114
5.96 Catch::Generators::GeneratorUntypedBase Class Reference	115
5.96.1 Member Function Documentation	115
5.96.1.1 countedNext()	116
5.96.1.2 currentElementAsString()	116
5.97 Catch::Generators::GeneratorWrapper< T > Class Template Reference	116
5.98 Catch::Detail::has_description< T, typename > Struct Template Reference	117
5.99 Catch::Detail::has_description< T, void_t< decltype(T::getDescription())> > Struct Template Reference	118
5.100 Catch::Matchers::HasSizeMatcher Class Reference	119
5.101 Catch::Clara::Help Struct Reference	120
5.102 Catch::Clara::Detail::HelpColumns Struct Reference	121
5.103 Catch::IConfig Class Reference	122

5.104 Catch::IContext Class Reference	123
5.105 Catch::IEventListener Class Reference	124
5.105.1 Detailed Description	126
5.105.2 Member Function Documentation	126
5.105.2.1 testRunEnded()	126
5.105.2.2 testRunStarting()	126
5.106 Catch::IExceptionTranslator Class Reference	127
5.107 Catch::IExceptionTranslatorRegistry Class Reference	127
5.108 Catch::Generators::IGenerator< T > Class Template Reference	128
5.109 Catch::IGeneratorTracker Class Reference	129
5.110 Catch::IMutableContext Class Reference	129
5.111 Catch::IMutableEnumValuesRegistry Class Reference	130
5.112 Catch::IMutableRegistryHub Class Reference	131
5.113 Catch::IRegistryHub Class Reference	131
5.114 Catch::IReporterFactory Class Reference	131
5.115 Catch::IReporterRegistry Class Reference	132
5.116 Catch::IResultCapture Class Reference	133
5.117 Catch::is_callable< T > Struct Template Reference	134
5.118 Catch::is_callable< Fun(Args...) > Struct Template Reference	134
5.119 Catch::is_callable_tester Struct Reference	135
5.120 Catch::is_range< T > Struct Template Reference	135
5.121 Catch::Detail::is_range_impl< T, typename > Struct Template Reference	136
5.122 Catch::Detail::is_range_impl< T, void_t< decltype(begin(std::declval< T >())) > > Struct Template Reference	137
5.123 Catch::Benchmark::Detail::is_related< T, U > Struct Template Reference	138
5.124 Catch::Clara::Detail::is_unary_function< F, typename > Struct Template Reference	139
5.125 Catch::Clara::Detail::is_unary_function< F, Catch::Detail::void_t< decltype(std::declval< F >())(fake_arg()) > > Struct Template Reference	140
5.126 Catch::Matchers::IsEmptyMatcher Class Reference	141
5.127 Catch::ISingleton Struct Reference	142
5.128 Catch::Detail::IsStreamInsertable< T > Class Template Reference	143
5.129 Catch::IStream Class Reference	143
5.129.1 Member Function Documentation	143
5.129.1.1 isConsole()	143
5.130 Catch::ITagAliasRegistry Class Reference	144
5.131 Catch::TextFlow::Columns::iterator Class Reference	144
5.132 Catch::Generators::IteratorGenerator< T > Class Template Reference	145
5.132.1 Member Function Documentation	146
5.132.1.1 next()	146
5.133 Catch::ITestCaseRegistry Class Reference	146
5.134 Catch::ITestInvoker Class Reference	147
5.135 Catch::TestCaseTracking::ITracker Class Reference	147
5.135.1 Member Function Documentation	149

5.135.1.1 findChild()	149
5.135.1.2 isGeneratorTracker()	149
5.135.1.3 isSectionTracker()	149
5.136 Catch::ITransientExpression Class Reference	149
5.137 Catch::JUnitReporter Class Reference	150
5.137.1 Member Function Documentation	151
5.137.1.1 testRunStarting()	151
5.138 Catch::Clara::Detail::LambdaInvoker< ReturnType > Struct Template Reference	151
5.139 Catch::Clara::Detail::LambdaInvoker< void > Struct Reference	152
5.140 Catch::LazyExpression Class Reference	152
5.141 Catch::LeakDetector Struct Reference	152
5.142 LFMCMC< TData > Class Template Reference	152
5.142.1 Detailed Description	153
5.143 Catch::lineOfChars Struct Reference	154
5.144 Catch::ListenerDescription Struct Reference	154
5.145 Catch::ListenerRegistrar< T > Class Template Reference	155
5.146 Catch::Detail::make_void<... > Struct Template Reference	155
5.147 Catch::Generators::MapGenerator< T, U, Func > Class Template Reference	155
5.147.1 Member Function Documentation	156
5.147.1.1 next()	156
5.148 Catch::Matchers::Detail::MatchAllOf< ArgT > Class Template Reference	157
5.149 Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs > Class Template Reference	158
5.150 Catch::Matchers::Detail::MatchAnyOf< ArgT > Class Template Reference	160
5.151 Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs > Class Template Reference	161
5.152 Catch::Matchers::MatcherBase< T > Class Template Reference	163
5.153 Catch::Matchers::MatcherGenericBase Class Reference	164
5.154 Catch::Matchers::MatcherUntypedBase Class Reference	166
5.155 Catch::MatchExpr< ArgT, MatcherT > Class Template Reference	167
5.156 Catch::Matchers::Detail::MatchNotOf< ArgT > Class Template Reference	168
5.157 Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT > Class Template Reference	170
5.158 Catch::MessageBuilder Struct Reference	171
5.159 Catch::MessageInfo Struct Reference	172
5.160 Catch::MessageStream Struct Reference	173
5.161 Model< TSeq > Class Template Reference	174
5.161.1 Detailed Description	180
5.161.2 Member Function Documentation	180
5.161.2.1 add_global_action()	180
5.161.2.2 reset()	181
5.161.2.3 run_multiple()	181
5.161.2.4 write_data()	181
5.162 Catch::MultiReporter Class Reference	182
5.162.1 Member Function Documentation	184



5.162.1.1 testRunEnded()	184
5.162.1.2 testRunStarting()	184
5.163 Catch::TestCaseTracking::NameAndLocation Struct Reference	185
5.164 Catch::NameAndTags Struct Reference	186
5.165 Catch::CumulativeReporterBase::Node< T, ChildNodeT > Struct Template Reference	186
5.166 Catch::Detail::NonCopyable Class Reference	187
5.166.1 Detailed Description	187
5.167 Catch::Matchers::NoneMatchMatcher< Matcher > Class Template Reference	188
5.168 Catch::Benchmark::now< Clock > Struct Template Reference	189
5.169 Catch::Benchmark::Detail::ObjectStorage< T, Destruct > Struct Template Reference	189
5.170 Catch::Clara::Opt Class Reference	190
5.171 Catch::Optional< T > Class Template Reference	192
5.172 Catch::Benchmark::OutlierClassification Struct Reference	192
5.173 Catch::Clara::Parser Class Reference	193
5.174 Catch::Clara::Detail::ParserBase Class Reference	194
5.175 Catch::Clara::Detail::ParserRefImpl< DerivedT > Class Template Reference	195
5.176 Catch::Clara::Detail::ParseState Class Reference	196
5.177 PersonTools< TSeq > Class Template Reference	196
5.178 Catch::pluralise Class Reference	197
5.178.1 Detailed Description	197
5.179 Catch::Matchers::PredicateMatcher< T, Predicate > Class Template Reference	197
5.180 Catch::ProcessedReporterSpec Struct Reference	198
5.180.1 Detailed Description	199
5.181 Progress Class Reference	199
5.181.1 Detailed Description	199
5.182 Queue< TSeq > Class Template Reference	199
5.182.1 Detailed Description	199
5.183 RandGraph Class Reference	200
5.184 Catch::Generators::RandomFloatingGenerator< Float > Class Template Reference	200
5.184.1 Member Function Documentation	201
5.184.1.1 next()	201
5.185 Catch::Generators::RandomIntegerGenerator< Integer > Class Template Reference	202
5.185.1 Member Function Documentation	203
5.185.1.1 next()	203
5.186 Catch::Generators::RangeGenerator< T > Class Template Reference	203
5.186.1 Member Function Documentation	204
5.186.1.1 next()	204
5.187 Catch::ratio_string< Ratio > Struct Template Reference	205
5.188 Catch::ratio_string< std::atto > Struct Reference	205
5.189 Catch::ratio_string< std::femto > Struct Reference	205
5.190 Catch::ratio_string< std::micro > Struct Reference	205
5.191 Catch::ratio_string< std::milli > Struct Reference	206

5.192 Catch::ratio_string< std::nano > Struct Reference . . . . .	206
5.193 Catch::ratio_string< std::pico > Struct Reference . . . . .	206
5.194 Catch::RedirectedStdErr Class Reference . . . . .	206
5.195 Catch::RedirectedStdOut Class Reference . . . . .	207
5.196 Catch::RedirectedStream Class Reference . . . . .	207
5.197 Catch::RedirectedStreams Class Reference . . . . .	207
5.198 Catch::Matchers::RegexMatcher Class Reference . . . . .	208
5.199 Catch::RegistrarForTagAliases Struct Reference . . . . .	209
5.200 Catch::Benchmark::Detail::repeater< Fun > Struct Template Reference . . . . .	209
5.201 Catch::Generators::RepeatGenerator< T > Class Template Reference . . . . .	210
5.201.1 Member Function Documentation . . . . .	211
5.201.1.1 next() . . . . .	211
5.202 Catch::ReporterBase Class Reference . . . . .	211
5.202.1 Detailed Description . . . . .	212
5.202.2 Member Function Documentation . . . . .	212
5.202.2.1 listListeners() . . . . .	212
5.202.2.2 listReporters() . . . . .	213
5.202.2.3 listTags() . . . . .	213
5.202.2.4 listTests() . . . . .	213
5.202.3 Member Data Documentation . . . . .	213
5.202.3.1 m_stream . . . . .	214
5.203 Catch::ReporterConfig Struct Reference . . . . .	214
5.204 Catch::ReporterDescription Struct Reference . . . . .	214
5.205 Catch::ReporterFactory< T > Class Template Reference . . . . .	215
5.206 Catch::ReporterPreferences Struct Reference . . . . .	215
5.206.1 Detailed Description . . . . .	216
5.206.2 Member Data Documentation . . . . .	216
5.206.2.1 shouldRedirectStdOut . . . . .	216
5.206.2.2 shouldReportAllAssertions . . . . .	216
5.207 Catch::ReporterRegistrar< T > Class Template Reference . . . . .	216
5.208 Catch::ReporterRegistry Class Reference . . . . .	217
5.209 Catch::ReporterSpec Class Reference . . . . .	218
5.209.1 Detailed Description . . . . .	218
5.210 Catch::Clara::Detail::ResultBase Class Reference . . . . .	218
5.211 Catch::ResultDisposition Struct Reference . . . . .	219
5.212 Catch::Clara::Detail::ResultValueBase< T > Class Template Reference . . . . .	219
5.213 Catch::Clara::Detail::ResultValueBase< void > Class Reference . . . . .	221
5.214 Catch::ResultWas Struct Reference . . . . .	222
5.215 Catch::ReusableStringStream Class Reference . . . . .	222
5.216 Catch::RunContext Class Reference . . . . .	223
5.217 Catch::Benchmark::SampleAnalysis< Duration > Struct Template Reference . . . . .	225
5.218 Catch::XmlWriter::ScopedElement Class Reference . . . . .	225

5.219	<a href="#">Catch::ScopedMessage Class Reference</a>	226
5.220	<a href="#">Catch::Section Class Reference</a>	227
5.221	<a href="#">Catch::SectionEndInfo Struct Reference</a>	228
5.222	<a href="#">Catch::SectionInfo Struct Reference</a>	228
5.223	<a href="#">Catch::CumulativeReporterBase::SectionNode Struct Reference</a>	229
5.224	<a href="#">Catch::SectionStats Struct Reference</a>	230
5.225	<a href="#">Catch::TestCaseTracking::SectionTracker Class Reference</a>	231
5.225.1	<a href="#">Member Function Documentation</a>	232
5.225.1.1	<a href="#">isSectionTracker()</a>	232
5.226	<a href="#">Catch::Session Class Reference</a>	232
5.227	<a href="#">Catch::SimplePcg32 Class Reference</a>	233
5.228	<a href="#">Catch::Singleton&lt; SingletonImplT, InterfaceT, MutableInterfaceT &gt; Class Template Reference</a>	234
5.229	<a href="#">Catch::Generators::SingleValueGenerator&lt; T &gt; Class Template Reference</a>	235
5.229.1	<a href="#">Member Function Documentation</a>	236
5.229.1.1	<a href="#">next()</a>	236
5.230	<a href="#">Catch::Matchers::SizeMatchesMatcher&lt; Matcher &gt; Class Template Reference</a>	237
5.231	<a href="#">Catch::SonarQubeReporter Class Reference</a>	238
5.231.1	<a href="#">Member Function Documentation</a>	239
5.231.1.1	<a href="#">testRunStarting()</a>	239
5.232	<a href="#">Catch::SourceLineInfo Struct Reference</a>	239
5.233	<a href="#">Catch::Matchers::StartsWithMatcher Class Reference</a>	240
5.234	<a href="#">Catch::StartupExceptionRegistry Class Reference</a>	241
5.235	<a href="#">Catch::StreamEndStop Struct Reference</a>	241
5.236	<a href="#">Catch::StreamingReporterBase Class Reference</a>	241
5.236.1	<a href="#">Member Function Documentation</a>	242
5.236.1.1	<a href="#">testRunEnded()</a>	243
5.236.1.2	<a href="#">testRunStarting()</a>	243
5.237	<a href="#">Catch::Matchers::StringContainsMatcher Class Reference</a>	244
5.238	<a href="#">Catch::Matchers::StringEqualsMatcher Class Reference</a>	245
5.239	<a href="#">Catch::StringMaker&lt; T, typename &gt; Struct Template Reference</a>	246
5.240	<a href="#">Catch::StringMaker&lt; bool &gt; Struct Reference</a>	246
5.241	<a href="#">Catch::StringMaker&lt; Catch::Approx &gt; Struct Reference</a>	246
5.242	<a href="#">Catch::StringMaker&lt; char * &gt; Struct Reference</a>	246
5.243	<a href="#">Catch::StringMaker&lt; char &gt; Struct Reference</a>	247
5.244	<a href="#">Catch::StringMaker&lt; char const * &gt; Struct Reference</a>	247
5.245	<a href="#">Catch::StringMaker&lt; char[SZ]&gt; Struct Template Reference</a>	247
5.246	<a href="#">Catch::StringMaker&lt; double &gt; Struct Reference</a>	247
5.247	<a href="#">Catch::StringMaker&lt; float &gt; Struct Reference</a>	248
5.248	<a href="#">Catch::StringMaker&lt; int &gt; Struct Reference</a>	248
5.249	<a href="#">Catch::StringMaker&lt; long &gt; Struct Reference</a>	248
5.250	<a href="#">Catch::StringMaker&lt; long long &gt; Struct Reference</a>	248
5.251	<a href="#">Catch::StringMaker&lt; R C::* &gt; Struct Template Reference</a>	249

5.252 Catch::StringMaker< R, std::enable_if_t< is_range< R >::value &&!::Catch::Detail::IsStream← Insertable< R >::value > > Struct Template Reference . . . . .	249
5.253 Catch::StringMaker< signed char > Struct Reference . . . . .	249
5.254 Catch::StringMaker< signed char[SZ]> Struct Template Reference . . . . .	249
5.255 Catch::StringMaker< std::chrono::duration< Value, Ratio > > Struct Template Reference . . . . .	250
5.256 Catch::StringMaker< std::chrono::duration< Value, std::ratio< 1 > > > Struct Template Reference . . . . .	250
5.257 Catch::StringMaker< std::chrono::duration< Value, std::ratio< 3600 > > > Struct Template Ref- erence . . . . .	250
5.258 Catch::StringMaker< std::chrono::duration< Value, std::ratio< 60 > > > Struct Template Reference . . . . .	250
5.259 Catch::StringMaker< std::chrono::time_point< Clock, Duration > > Struct Template Reference . . . . .	251
5.260 Catch::StringMaker< std::chrono::time_point< std::chrono::system_clock, Duration > > Struct Template Reference . . . . .	251
5.261 Catch::StringMaker< std::nullptr_t > Struct Reference . . . . .	251
5.262 Catch::StringMaker< std::string > Struct Reference . . . . .	251
5.263 Catch::StringMaker< std::wstring > Struct Reference . . . . .	252
5.264 Catch::StringMaker< T * > Struct Template Reference . . . . .	252
5.265 Catch::StringMaker< T[SZ]> Struct Template Reference . . . . .	252
5.266 Catch::StringMaker< unsigned char > Struct Reference . . . . .	252
5.267 Catch::StringMaker< unsigned char[SZ]> Struct Template Reference . . . . .	253
5.268 Catch::StringMaker< unsigned int > Struct Reference . . . . .	253
5.269 Catch::StringMaker< unsigned long > Struct Reference . . . . .	253
5.270 Catch::StringMaker< unsigned long long > Struct Reference . . . . .	253
5.271 Catch::StringMaker< wchar_t * > Struct Reference . . . . .	254
5.272 Catch::StringMaker< wchar_t const * > Struct Reference . . . . .	254
5.273 Catch::Matchers::StringMatcherBase Class Reference . . . . .	254
5.274 Catch::StringRef Class Reference . . . . .	255
5.274.1 Detailed Description . . . . .	256
5.274.2 Member Function Documentation . . . . .	256
5.274.2.1 compare() . . . . .	256
5.275 Catch::Tag Struct Reference . . . . .	256
5.275.1 Detailed Description . . . . .	257
5.276 Catch::TagAlias Struct Reference . . . . .	257
5.277 Catch::TagAliasRegistry Class Reference . . . . .	258
5.278 Catch::TagInfo Struct Reference . . . . .	258
5.279 Catch::Generators::TakeGenerator< T > Class Template Reference . . . . .	259
5.279.1 Member Function Documentation . . . . .	260
5.279.1.1 next() . . . . .	260
5.280 Catch::TAPReporter Class Reference . . . . .	261
5.280.1 Member Function Documentation . . . . .	262
5.280.1.1 testRunEnded() . . . . .	262
5.280.1.2 testRunStarting() . . . . .	262
5.281 Catch::TeamCityReporter Class Reference . . . . .	263
5.281.1 Member Function Documentation . . . . .	264

5.281.1.1 testRunEnded()	264
5.281.1.2 testRunStarting()	264
5.282 Catch::TestCaseHandle Class Reference	264
5.282.1 Detailed Description	265
5.283 Catch::TestCaseInfo Struct Reference	265
5.283.1 Detailed Description	266
5.284 Catch::TestCaseInfoHasher Class Reference	266
5.285 Catch::TestCaseStats Struct Reference	267
5.286 Catch::TestFailureException Struct Reference	267
5.286.1 Detailed Description	267
5.287 Catch::TestInvokerAsFunction Class Reference	268
5.288 Catch::TestInvokerAsMethod< C > Class Template Reference	269
5.289 Catch::TestRegistry Class Reference	270
5.290 Catch::TestRunInfo Struct Reference	271
5.291 Catch::TestRunStats Struct Reference	271
5.292 Catch::TestSpec Class Reference	272
5.293 Catch::TestSpecParser Class Reference	273
5.294 Catch::Timer Class Reference	273
5.295 Catch::Benchmark::Timing< Duration, Result > Struct Template Reference	273
5.296 Catch::Clara::Detail::Token Struct Reference	273
5.297 Catch::Clara::Detail::TokenStream Class Reference	274
5.298 Tool< TSeq > Class Template Reference	274
5.298.1 Detailed Description	275
5.299 Tools< TSeq > Class Template Reference	275
5.299.1 Detailed Description	276
5.300 Tools_const< TSeq > Class Template Reference	276
5.300.1 Detailed Description	277
5.301 Catch::Totals Struct Reference	277
5.302 Catch::TestCaseTracking::TrackerBase Class Reference	278
5.303 Catch::TestCaseTracking::TrackerContext Class Reference	279
5.304 Catch::true_given< typename > Struct Template Reference	280
5.305 Catch::Benchmark::Detail::CompleteType< void >::type Struct Reference	280
5.306 Catch::UnaryExpr< LhsT > Class Template Reference	281
5.307 Catch::Clara::Detail::UnaryLambdaTraits< L > Struct Template Reference	281
5.308 Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::*)(Args...) const > Struct Template Reference	282
5.309 Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::*)(ArgT) const > Struct Template Reference	282
5.310 Catch::Detail::unique_ptr< T > Class Template Reference	282
5.310.1 Detailed Description	283
5.311 Catch::Matchers::UnorderedEqualsMatcher< T, AllocComp, AllocMatch > Class Template Reference	283
5.312 UserData< TSeq > Class Template Reference	284
5.312.1 Detailed Description	285

5.312.2 Constructor & Destructor Documentation	286
5.312.2.1 UserData()	286
5.313 vecHasher< T > Struct Template Reference	286
5.313.1 Detailed Description	286
5.314 Catch::Matchers::VectorContainsElementMatcher< T, Alloc > Class Template Reference	287
5.315 Catch::Version Struct Reference	288
5.316 Virus< TSeq > Class Template Reference	289
5.316.1 Detailed Description	290
5.317 Viruses< TSeq > Class Template Reference	291
5.317.1 Detailed Description	291
5.318 Viruses_const< TSeq > Class Template Reference	291
5.318.1 Detailed Description	292
5.319 Catch::WaitForKeypress Struct Reference	292
5.320 Catch::WarnAbout Struct Reference	292
5.320.1 Member Enumeration Documentation	293
5.320.1.1 What	293
5.321 Catch::WildcardPattern Class Reference	293
5.322 Catch::Matchers::WithinAbsMatcher Class Reference	293
5.323 Catch::Matchers::WithinRelMatcher Class Reference	295
5.324 Catch::Matchers::WithinUlpMatcher Class Reference	296
5.325 Catch::XmlEncode Class Reference	297
5.325.1 Detailed Description	297
5.326 Catch::XmlReporter Class Reference	298
5.326.1 Member Function Documentation	299
5.326.1.1 listListeners()	299
5.326.1.2 listReporters()	299
5.326.1.3 listTags()	300
5.326.1.4 listTests()	300
5.326.1.5 testRunEnded()	300
5.326.1.6 testRunStarting()	300
5.327 Catch::XmlWriter Class Reference	301
5.327.1 Member Function Documentation	301
5.327.1.1 writeAttribute()	301
<b>6 File Documentation</b>	<b>303</b>
6.1 include/catch2/catch_amalgamated.hpp File Reference	303
6.1.1 Detailed Description	330
6.1.2 Macro Definition Documentation	331
6.1.2.1 CATCH_INTERNAL_DEFINE_EXPRESSION_OPERATOR	331
6.1.2.2 CATCH_REGISTER_LISTENER	331
6.1.2.3 CATCH_REGISTER_REPORTER	331
6.1.2.4 CATCH_REGISTER_TAG_ALIAS	332

6.1.2.5 GENERATE . . . . .	332
6.1.2.6 GENERATE_COPY . . . . .	332
6.1.2.7 GENERATE_REF . . . . .	332
6.1.2.8 INTERNAL_CATCH_BENCHMARK . . . . .	333
6.1.2.9 INTERNAL_CATCH_BENCHMARK_ADVANCED . . . . .	333
6.1.2.10 INTERNAL_CATCH_CAPTURE . . . . .	333
6.1.2.11 INTERNAL_CATCH_DECLARE_SIG_TEST1 . . . . .	333
6.1.2.12 INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD1 . . . . .	334
6.1.2.13 INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD_X . . . . .	334
6.1.2.14 INTERNAL_CATCH_DECLARE_SIG_TEST_X . . . . .	334
6.1.2.15 INTERNAL_CATCH_DEFINE_SIG_TEST1 . . . . .	334
6.1.2.16 INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD1 . . . . .	335
6.1.2.17 INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD_X . . . . .	335
6.1.2.18 INTERNAL_CATCH_DEFINE_SIG_TEST_X . . . . .	335
6.1.2.19 INTERNAL_CATCH_DYNAMIC_SECTION . . . . .	335
6.1.2.20 INTERNAL_CATCH_ELSE . . . . .	336
6.1.2.21 INTERNAL_CATCH_IF . . . . .	336
6.1.2.22 INTERNAL_CATCH_METHOD_AS_TEST_CASE . . . . .	336
6.1.2.23 INTERNAL_CATCH_MSG . . . . .	336
6.1.2.24 INTERNAL_CATCH_NO_THROW . . . . .	337
6.1.2.25 INTERNAL_CATCH_NTTP_1 . . . . .	337
6.1.2.26 INTERNAL_CATCH_NTTP_REGISTER . . . . .	337
6.1.2.27 INTERNAL_CATCH_NTTP_REGISTER0 . . . . .	338
6.1.2.28 INTERNAL_CATCH_NTTP_REGISTER_METHOD . . . . .	338
6.1.2.29 INTERNAL_CATCH_NTTP_REGISTER_METHOD0 . . . . .	338
6.1.2.30 INTERNAL_CATCH_REGISTER_ENUM . . . . .	338
6.1.2.31 INTERNAL_CATCH_REGISTER_TESTCASE . . . . .	339
6.1.2.32 INTERNAL_CATCH_SECTION . . . . .	339
6.1.2.33 INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_2 . . . . .	339
6.1.2.34 INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD_2 . . . . .	340
6.1.2.35 INTERNAL_CATCH_TEMPLATE_TEST_CASE_2 . . . . .	340
6.1.2.36 INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD_2 . . . . .	341
6.1.2.37 INTERNAL_CATCH_TEST . . . . .	341
6.1.2.38 INTERNAL_CATCH_TEST_CASE_METHOD2 . . . . .	342
6.1.2.39 INTERNAL_CATCH_TESTCASE2 . . . . .	342
6.1.2.40 INTERNAL_CATCH_THROWS . . . . .	343
6.1.2.41 INTERNAL_CATCH_THROWS_AS . . . . .	343
6.1.2.42 INTERNAL_CATCH_THROWS_MATCHES . . . . .	343
6.1.2.43 INTERNAL_CATCH_THROWS_STR_MATCHES . . . . .	344
6.1.2.44 INTERNAL_CATCH_TRANSLATE_EXCEPTION2 . . . . .	344
6.1.2.45 INTERNAL_CHECK_THAT . . . . .	345
6.1.3 Enumeration Type Documentation . . . . .	345

6.1.3.1 ColourMode . . . . .	345
6.1.3.2 GenerateFrom . . . . .	345
6.1.3.3 ResultType . . . . .	345
6.1.4 Function Documentation . . . . .	346
6.1.4.1 Contains() [1/2] . . . . .	346
6.1.4.2 Contains() [2/2] . . . . .	346
6.1.4.3 convertIntoString() . . . . .	346
6.1.4.4 defaultListListeners() . . . . .	346
6.1.4.5 defaultListReporters() . . . . .	347
6.1.4.6 defaultListTags() . . . . .	347
6.1.4.7 defaultListTests() . . . . .	347
6.1.4.8 makeStream() . . . . .	347
6.1.4.9 operator&&() [1/2] . . . . .	348
6.1.4.10 operator&&() [2/2] . . . . .	348
6.1.4.11 operator"   "   () [1/2] . . . . .	348
6.1.4.12 operator"   "   () [2/2] . . . . .	348
6.1.4.13 parseReporterSpec() . . . . .	349
6.1.4.14 Predicate() . . . . .	349
6.1.4.15 registerReporterImpl() . . . . .	349
6.1.4.16 ulpDistance() . . . . .	349
<b>Index</b>	<b>351</b>



# Chapter 1

## Source code

Although `epiworld` is a header-only C++ library, we do make use of other cool projects in some of the examples and for testing. The `cxxopts` library provides a smooth interface for using options in your C++ program. `catch2` is a great C++ unit-tests framework that we have use in [other projects](#).

Although you are free to use `epiworld/`, we also have a single-header version that incorporates all its components [here](#).



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Catch::Clara::accept_many_t . . . . .	21
Action< TSeq > . . . . .	21
AdjList . . . . .	23
Agent< TSeq > . . . . .	24
Agent< int > . . . . .	24
Agent< TSeq > . . . . .	24
Catch::Approx . . . . .	30
Catch::Clara::Args . . . . .	33
Catch::Generators::as< T > . . . . .	34
Catch::AssertionHandler . . . . .	34
Catch::AssertionInfo . . . . .	34
Catch::Detail::AssertionOrBenchmarkResult . . . . .	35
Catch::AssertionReaction . . . . .	35
Catch::AssertionResult . . . . .	36
Catch::AssertionResultData . . . . .	37
Catch::AssertionStats . . . . .	38
Catch::Benchmark::Benchmark . . . . .	42
Catch::Benchmark::Detail::BenchmarkFunction . . . . .	42
Catch::BenchmarkInfo . . . . .	43
Catch::BenchmarkStats< Duration > . . . . .	44
Catch::Benchmark::Detail::bootstrap_analysis . . . . .	46
Catch::Capturer . . . . .	61
Catch::Matchers::CasedString . . . . .	62
Catch::Detail::CaseInsensitiveEqualTo . . . . .	62
Catch::Detail::CaseInsensitiveLess . . . . .	62
Catch_global_namespace_dummy . . . . .	63
Catch::Benchmark::Chronometer . . . . .	63
Catch::Benchmark::Detail::ChronometerConcept . . . . .	63
Catch::Benchmark::Detail::ChronometerModel< Clock > . . . . .	64
Catch::Colour . . . . .	67
Catch::ColourImpl::ColourGuard . . . . .	67
Catch::ColourImpl . . . . .	68
Catch::TextFlow::Column . . . . .	69
Catch::TextFlow::Columns . . . . .	70
Catch::Benchmark::Detail::CompleterInvoker< Result > . . . . .	72

Catch::Benchmark::Detail::CompleteInvoker< void > . . . . .	73
Catch::Benchmark::Detail::CompleteType< T > . . . . .	73
Catch::Benchmark::Detail::CompleteType< void > . . . . .	73
Catch::ConfigData . . . . .	76
Catch::TextFlow::Column::const_iterator . . . . .	81
Catch::Counts . . . . .	86
DataBase< TSeq > . . . . .	90
DataBase< int > . . . . .	90
Catch::Decomposer . . . . .	92
Entity< TSeq > . . . . .	94
Catch::Detail::EnumInfo . . . . .	94
Catch::Benchmark::Environment< Clock > . . . . .	96
Catch::Benchmark::EnvironmentEstimate< Duration > . . . . .	97
Catch::Benchmark::EnvironmentEstimate< FloatDuration< Clock > > . . . . .	97
Catch::ErrnoGuard . . . . .	99
Catch::Benchmark::Estimate< Duration > . . . . .	99
Catch::Benchmark::Estimate< double > . . . . .	99
Catch::EventListenerFactory . . . . .	102
std::exception	
Catch::GeneratorException . . . . .	112
Catch::ExceptionTranslatorRegistrar . . . . .	104
Catch::Benchmark::ExecutionPlan< Duration > . . . . .	105
Catch::ExprLhs< LhsT > . . . . .	107
Catch::Clara::Detail::fake_arg . . . . .	107
std::false_type	
Catch::Clara::Detail::is_unary_function< F, typename > . . . . .	139
Catch::Detail::has_description< T, typename > . . . . .	117
Catch::Detail::is_range_impl< T, typename > . . . . .	136
Catch::is_range< T > . . . . .	135
Catch::always_false< T > . . . . .	28
Catch::FatalConditionHandler . . . . .	108
Catch::FatalConditionHandlerGuard . . . . .	108
Catch::TestSpec::FilterMatch . . . . .	110
Catch::Generators::GeneratorUntypedBase . . . . .	115
Catch::Generators::IGenerator< Float > . . . . .	128
Catch::Generators::RandomFloatingGenerator< Float > . . . . .	200
Catch::Generators::IGenerator< std::vector< T > > . . . . .	128
Catch::Generators::ChunkGenerator< T > . . . . .	65
Catch::Generators::IGenerator< Integer > . . . . .	128
Catch::Generators::RandomIntegerGenerator< Integer > . . . . .	202
Catch::Generators::IGenerator< T > . . . . .	128
Catch::Generators::FilterGenerator< T, Predicate > . . . . .	109
Catch::Generators::FixedValuesGenerator< T > . . . . .	111
Catch::Generators::Generators< T > . . . . .	113
Catch::Generators::IteratorGenerator< T > . . . . .	145
Catch::Generators::MapGenerator< T, U, Func > . . . . .	155
Catch::Generators::RangeGenerator< T > . . . . .	203
Catch::Generators::RepeatGenerator< T > . . . . .	210
Catch::Generators::SingleValueGenerator< T > . . . . .	235
Catch::Generators::TakeGenerator< T > . . . . .	259
Catch::Generators::GeneratorWrapper< T > . . . . .	116
Catch::Generators::GeneratorWrapper< U > . . . . .	116
Catch::Clara::Detail::HelpColumns . . . . .	121
Catch::IContext . . . . .	123
Catch::IMutableContext . . . . .	129
Catch::IEventListener . . . . .	124
Catch::EventListenerBase . . . . .	100

Catch::MultiReporter	182
Catch::ReporterBase	211
Catch::CumulativeReporterBase	87
Catch::JUnitReporter	150
Catch::SonarQubeReporter	238
Catch::StreamingReporterBase	241
Catch::AutomakeReporter	39
Catch::CompactReporter	71
Catch::ConsoleReporter	79
Catch::TAPReporter	261
Catch::TeamCityReporter	263
Catch::XmlReporter	298
Catch::IExceptionTranslator	127
Catch::IExceptionTranslatorRegistry	127
Catch::ExceptionTranslatorRegistry	104
Catch::IGeneratorTracker	129
Catch::IMutableEnumValuesRegistry	130
Catch::Detail::EnumValuesRegistry	95
Catch::IMutableRegistryHub	131
std::integral_constant	
Catch::Matchers::Detail::conjunction< Cond, Rest... >	78
Catch::IRegistryHub	131
Catch::IReporterFactory	131
Catch::ReporterFactory< T >	215
Catch::IReporterRegistry	132
Catch::ReporterRegistry	217
Catch::IResultCapture	133
Catch::RunContext	223
Catch::is_callable< T >	134
Catch::is_callable_tester	135
std::is_same	
Catch::Benchmark::Detail::is_related< T, U >	138
Catch::ISingleton	142
Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT >	234
Catch::Detail::IsStreamInsertable< T >	143
Catch::IStream	143
Catch::ITagAliasRegistry	144
Catch::TagAliasRegistry	258
Catch::TextFlow::Columns::iterator	144
Catch::ITestCaseRegistry	146
Catch::TestRegistry	270
Catch::ITestInvoker	147
Catch::TestInvokerAsFunction	268
Catch::TestInvokerAsMethod< C >	269
Catch::TestCaseTracking::ITracker	147
Catch::TestCaseTracking::TrackerBase	278
Catch::TestCaseTracking::SectionTracker	231
Catch::ITransientExpression	149
Catch::BinaryExpr< LhsT, RhsT >	45
Catch::MatchExpr< ArgT, MatcherT >	167
Catch::UnaryExpr< LhsT >	281
Catch::Clara::Detail::LambdaInvoker< ReturnType >	151
Catch::Clara::Detail::LambdaInvoker< void >	152
Catch::LazyExpression	152
Catch::LeakDetector	152

LFMCMC< TData > . . . . .	152
Catch::lineOfChars . . . . .	154
Catch::ListenerDescription . . . . .	154
Catch::ListenerRegistrar< T > . . . . .	155
Catch::Detail::make_void<... > . . . . .	155
Catch::Matchers::MatcherUntypedBase . . . . .	166
Catch::Matchers::MatcherBase< double > . . . . .	163
Catch::Matchers::WithinAbsMatcher . . . . .	293
Catch::Matchers::WithinRelMatcher . . . . .	295
Catch::Matchers::WithinUlpsMatcher . . . . .	296
Catch::Matchers::MatcherBase< std::string > . . . . .	163
Catch::Matchers::RegexMatcher . . . . .	208
Catch::Matchers::StringMatcherBase . . . . .	254
Catch::Matchers::EndsWithMatcher . . . . .	93
Catch::Matchers::StartsWithMatcher . . . . .	240
Catch::Matchers::StringContainsMatcher . . . . .	244
Catch::Matchers::StringEqualsMatcher . . . . .	245
Catch::Matchers::MatcherBase< std::vector< T, AllocMatch > > . . . . .	163
Catch::Matchers::ApproxMatcher< T, AllocComp, AllocMatch > . . . . .	31
Catch::Matchers::ContainsMatcher< T, AllocComp, AllocMatch > . . . . .	84
Catch::Matchers::EqualsMatcher< T, AllocComp, AllocMatch > . . . . .	98
Catch::Matchers::UnorderedEqualsMatcher< T, AllocComp, AllocMatch > . . . . .	283
Catch::Matchers::MatcherBase< std::vector< T, Alloc > > . . . . .	163
Catch::Matchers::VectorContainsElementMatcher< T, Alloc > . . . . .	287
Catch::Matchers::MatcherBase< ArgT > . . . . .	163
Catch::Matchers::Detail::MatchAllOf< ArgT > . . . . .	157
Catch::Matchers::Detail::MatchAnyOf< ArgT > . . . . .	160
Catch::Matchers::Detail::MatchNotOf< ArgT > . . . . .	168
Catch::Matchers::MatcherBase< std::exception > . . . . .	163
Catch::Matchers::ExceptionMessageMatcher . . . . .	103
Catch::Matchers::MatcherBase< T > . . . . .	163
Catch::Matchers::PredicateMatcher< T, Predicate > . . . . .	197
Catch::Matchers::MatcherGenericBase . . . . .	164
Catch::Matchers::AllMatchMatcher< Matcher > . . . . .	27
Catch::Matchers::AnyMatchMatcher< Matcher > . . . . .	29
Catch::Matchers::ContainsElementMatcher< T, Equality > . . . . .	82
Catch::Matchers::ContainsMatcherMatcher< Matcher > . . . . .	85
Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs > . . . . .	158
Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs > . . . . .	161
Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT > . . . . .	170
Catch::Matchers::HasSizeMatcher . . . . .	119
Catch::Matchers::IsEmptyMatcher . . . . .	141
Catch::Matchers::NoneMatchMatcher< Matcher > . . . . .	188
Catch::Matchers::SizeMatchesMatcher< Matcher > . . . . .	237
Catch::MessageInfo . . . . .	172
Catch::MessageStream . . . . .	173
Catch::MessageBuilder . . . . .	171
Model< TSeq > . . . . .	174
Model< int > . . . . .	174
Model< TSeq > . . . . .	174
Catch::TestCaseTracking::NameAndLocation . . . . .	185
Catch::NameAndTags . . . . .	186
Catch::CumulativeReporterBase::Node< T, ChildNodeT > . . . . .	186
Catch::Detail::NonCopyable . . . . .	187
Catch::AutoReg . . . . .	40
Catch::Clara::Detail::BoundRef . . . . .	56
Catch::Clara::Detail::BoundFlagRefBase . . . . .	51

Catch::Clara::Detail::BoundFlagLambda< L > . . . . .	47
Catch::Clara::Detail::BoundFlagRef . . . . .	49
Catch::Clara::Detail::BoundValueRefBase . . . . .	60
Catch::Clara::Detail::BoundLambda< L > . . . . .	52
Catch::Clara::Detail::BoundManyLambda< L > . . . . .	54
Catch::Clara::Detail::BoundValueRef< T > . . . . .	57
Catch::Clara::Detail::BoundValueRef< std::vector< T > > . . . . .	59
Catch::IConfig . . . . .	122
Catch::Config . . . . .	75
Catch::ReusableStringStream . . . . .	222
Catch::Section . . . . .	227
Catch::Session . . . . .	232
Catch::TestCaseInfo . . . . .	265
Catch::Benchmark::now< Clock > . . . . .	189
Catch::Benchmark::Detail::ObjectStorage< T, Destruct > . . . . .	189
Catch::Optional< T > . . . . .	192
Catch::Optional< Catch::AssertionResult > . . . . .	192
Catch::Optional< Catch::AssertionStats > . . . . .	192
Catch::Optional< Catch::BenchmarkStats<> > . . . . .	192
Catch::Optional< ColourMode > . . . . .	192
Catch::Optional< std::string > . . . . .	192
Catch::Benchmark::OutlierClassification . . . . .	192
Catch::Clara::Detail::ParserBase . . . . .	194
Catch::Clara::Detail::ComposableParserImpl< ExeName > . . . . .	74
Catch::Clara::ExeName . . . . .	106
Catch::Clara::Detail::ComposableParserImpl< DerivedT > . . . . .	74
Catch::Clara::Detail::ParserRefImpl< Opt > . . . . .	195
Catch::Clara::Opt . . . . .	190
Catch::Clara::Help . . . . .	120
Catch::Clara::Detail::ParserRefImpl< Arg > . . . . .	195
Catch::Clara::Arg . . . . .	32
Catch::Clara::Detail::ParserRefImpl< DerivedT > . . . . .	195
Catch::Clara::Parser . . . . .	193
Catch::Clara::Detail::ParseState . . . . .	196
PersonTools< TSeq > . . . . .	196
Catch::pluralise . . . . .	197
Catch::ProcessedReporterSpec . . . . .	198
Progress . . . . .	199
Queue< TSeq > . . . . .	199
Queue< int > . . . . .	199
RandGraph . . . . .	200
Catch::ratio_string< Ratio > . . . . .	205
Catch::ratio_string< std::atto > . . . . .	205
Catch::ratio_string< std::femto > . . . . .	205
Catch::ratio_string< std::micro > . . . . .	205
Catch::ratio_string< std::milli > . . . . .	206
Catch::ratio_string< std::nano > . . . . .	206
Catch::ratio_string< std::pico > . . . . .	206
Catch::RedirectedStdErr . . . . .	206
Catch::RedirectedStdOut . . . . .	207
Catch::RedirectedStream . . . . .	207
Catch::RedirectedStreams . . . . .	207
Catch::RegistrarForTagAliases . . . . .	209
Catch::Benchmark::Detail::repeater< Fun > . . . . .	209
Catch::ReporterConfig . . . . .	214
Catch::ReporterDescription . . . . .	214
Catch::ReporterPreferences . . . . .	215

Catch::ReporterRegistrar< T > . . . . .	216
Catch::ReporterSpec . . . . .	218
Catch::Clara::Detail::ResultBase . . . . .	218
Catch::Clara::Detail::ResultValueBase< T > . . . . .	219
Catch::Clara::Detail::ResultValueBase< void > . . . . .	221
Catch::Clara::Detail::BasicResult< T > . . . . .	41
Catch::ResultDisposition . . . . .	219
Catch::ResultWas . . . . .	222
Catch::Benchmark::SampleAnalysis< Duration > . . . . .	225
Catch::XmlWriter::ScopedElement . . . . .	225
Catch::ScopedMessage . . . . .	226
Catch::SectionEndInfo . . . . .	228
Catch::SectionInfo . . . . .	228
Catch::CumulativeReporterBase::SectionNode . . . . .	229
Catch::SectionStats . . . . .	230
Catch::SimplePcg32 . . . . .	233
SingletonImplT	
Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT > . . . . .	234
Catch::SourceLineInfo . . . . .	239
Catch::StartupExceptionRegistry . . . . .	241
Catch::StreamEndStop . . . . .	241
Catch::StringMaker< T, typename > . . . . .	246
Catch::StringMaker< bool > . . . . .	246
Catch::StringMaker< Catch::Approx > . . . . .	246
Catch::StringMaker< char * > . . . . .	246
Catch::StringMaker< char > . . . . .	247
Catch::StringMaker< char const * > . . . . .	247
Catch::StringMaker< char[SZ]> . . . . .	247
Catch::StringMaker< double > . . . . .	247
Catch::StringMaker< float > . . . . .	248
Catch::StringMaker< int > . . . . .	248
Catch::StringMaker< long > . . . . .	248
Catch::StringMaker< long long > . . . . .	248
Catch::StringMaker< R C::* > . . . . .	249
Catch::StringMaker< R, std::enable_if_t< is_range< R >::value &&!::Catch::Detail::IsStreamInsertable< R >::value >> . . . . .	249
Catch::StringMaker< signed char > . . . . .	249
Catch::StringMaker< signed char[SZ]> . . . . .	249
Catch::StringMaker< std::chrono::duration< Value, Ratio > > . . . . .	250
Catch::StringMaker< std::chrono::duration< Value, std::ratio< 1 > > > . . . . .	250
Catch::StringMaker< std::chrono::duration< Value, std::ratio< 3600 > > > . . . . .	250
Catch::StringMaker< std::chrono::duration< Value, std::ratio< 60 > > > . . . . .	250
Catch::StringMaker< std::chrono::time_point< Clock, Duration > > . . . . .	251
Catch::StringMaker< std::chrono::time_point< std::chrono::system_clock, Duration > > . . . . .	251
Catch::StringMaker< std::nullptr_t > . . . . .	251
Catch::StringMaker< std::string > . . . . .	251
Catch::StringMaker< std::wstring > . . . . .	252
Catch::StringMaker< T * > . . . . .	252
Catch::StringMaker< T[SZ]> . . . . .	252
Catch::StringMaker< unsigned char > . . . . .	252
Catch::StringMaker< unsigned char[SZ]> . . . . .	253
Catch::StringMaker< unsigned int > . . . . .	253
Catch::StringMaker< unsigned long > . . . . .	253
Catch::StringMaker< unsigned long long > . . . . .	253
Catch::StringMaker< wchar_t * > . . . . .	254
Catch::StringMaker< wchar_t const * > . . . . .	254
Catch::StringRef . . . . .	255
Catch::Tag . . . . .	256



Catch::TagAlias	257
Catch::TagInfo	258
decltypeis_callable_tester::test	
Catch::is_callable< Fun(Args...) >	134
Catch::TestCaseHandle	264
Catch::TestCaseInfoHasher	266
Catch::TestCaseStats	267
Catch::TestFailureException	267
Catch::TestRunInfo	271
Catch::TestRunStats	271
Catch::TestSpec	272
Catch::TestSpecParser	273
Catch::Timer	273
Catch::Benchmark::Timing< Duration, Result >	273
Catch::Clara::Detail::Token	273
Catch::Clara::Detail::TokenStream	274
Tool< TSeq >	274
Tools< TSeq >	275
Tools_const< TSeq >	276
Catch::Totals	277
Catch::TestCaseTracking::TrackerContext	279
std::true_type	
Catch::Clara::Detail::is_unary_function< F, Catch::Detail::void_t< decltype(std::declval< F >())(fake← _arg()) > >	140
Catch::Detail::has_description< T, void_t< decltype(T::getDescription()) > >	118
Catch::Detail::is_range_impl< T, void_t< decltype(begin(std::declval< T >())) > >	137
Catch::Matchers::Detail::conjunction< Cond >	77
Catch::true_given< typename >	280
Catch::Benchmark::Detail::CompleteType< void >::type	280
Catch::Clara::Detail::UnaryLambdaTraits< L >	281
Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::*)(Args...) const >	282
Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::*)(ArgT) const >	282
Catch::Detail::unique_ptr< T >	282
Catch::Detail::unique_ptr< callable >	282
Catch::Detail::unique_ptr< Catch::ColourImpl >	282
Catch::Detail::unique_ptr< Catch::Config >	282
Catch::Detail::unique_ptr< Catch::CumulativeReporterBase::Node >	282
Catch::Detail::unique_ptr< Catch::CumulativeReporterBase::SectionNode >	282
Catch::Detail::unique_ptr< Catch::IStream >	282
Catch::Detail::unique_ptr< IEventListener >	282
Catch::Detail::unique_ptr< ITracker >	282
Catch::Detail::unique_ptr< TablePrinter >	282
UserData< TSeq >	284
vecHasher< T >	286
Catch::Version	288
Virus< TSeq >	289
Viruses< TSeq >	291
Viruses_const< TSeq >	291
Catch::WaitForKeypress	292
Catch::WarnAbout	292
Catch::WildcardPattern	293
Catch::XmlEncode	297
Catch::XmlWriter	301



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Catch::Clara::accept_many_t	21
Action< TSeq >	
Action data for update an agent	21
AdjList	23
Agent< TSeq >	
Agent (agents)	24
Catch::Matchers::AllMatchMatcher< Matcher >	27
Catch::always_false< T >	28
Catch::Matchers::AnyMatchMatcher< Matcher >	29
Catch::Approx	30
Catch::Matchers::ApproxMatcher< T, AllocComp, AllocMatch >	31
Catch::Clara::Arg	32
Catch::Clara::Args	33
Catch::Generators::as< T >	34
Catch::AssertionHandler	34
Catch::AssertionInfo	34
Catch::Detail::AssertionOrBenchmarkResult	
Represents either an assertion or a benchmark result to be handled by cumulative reporter later	35
Catch::AssertionReaction	35
Catch::AssertionResult	36
Catch::AssertionResultData	37
Catch::AssertionStats	38
Catch::AutomakeReporter	39
Catch::AutoReg	40
Catch::Clara::Detail::BasicResult< T >	41
Catch::Benchmark::Benchmark	42
Catch::Benchmark::Detail::BenchmarkFunction	42
Catch::BenchmarkInfo	43
Catch::BenchmarkStats< Duration >	44
Catch::BinaryExpr< LhsT, RhsT >	45
Catch::Benchmark::Detail::bootstrap_analysis	46
Catch::Clara::Detail::BoundFlagLambda< L >	47
Catch::Clara::Detail::BoundFlagRef	49
Catch::Clara::Detail::BoundFlagRefBase	51
Catch::Clara::Detail::BoundLambda< L >	52

Catch::Clara::Detail::BoundManyLambda< L > . . . . .	54
Catch::Clara::Detail::BoundRef . . . . .	56
Catch::Clara::Detail::BoundValueRef< T > . . . . .	57
Catch::Clara::Detail::BoundValueRef< std::vector< T > > . . . . .	59
Catch::Clara::Detail::BoundValueRefBase . . . . .	60
Catch::Catcher . . . . .	61
Catch::Matchers::CasedString . . . . .	62
Catch::Detail::CaseInsensitiveEqualTo . . . . .	
Provides case-insensitive op== semantics when called . . . . .	62
Catch::Detail::CaseInsensitiveLess . . . . .	
Provides case-insensitive op< semantics when called . . . . .	62
Catch_global_namespace_dummy . . . . .	63
Catch::Benchmark::Chronometer . . . . .	63
Catch::Benchmark::Detail::ChronometerConcept . . . . .	63
Catch::Benchmark::Detail::ChronometerModel< Clock > . . . . .	64
Catch::Generators::ChunkGenerator< T > . . . . .	65
Catch::Colour . . . . .	67
Catch::ColourImpl::ColourGuard . . . . .	67
Catch::ColourImpl . . . . .	68
Catch::TextFlow::Column . . . . .	69
Catch::TextFlow::Columns . . . . .	70
Catch::CompactReporter . . . . .	71
Catch::Benchmark::Detail::CompleterInvoker< Result > . . . . .	72
Catch::Benchmark::Detail::CompleterInvoker< void > . . . . .	73
Catch::Benchmark::Detail::CompleteType< T > . . . . .	73
Catch::Benchmark::Detail::CompleteType< void > . . . . .	73
Catch::Clara::Detail::ComposableParserImpl< DerivedT > . . . . .	74
Catch::Config . . . . .	75
Catch::ConfigData . . . . .	76
Catch::Matchers::Detail::conjunction< Cond > . . . . .	77
Catch::Matchers::Detail::conjunction< Cond, Rest... > . . . . .	78
Catch::ConsoleReporter . . . . .	79
Catch::TextFlow::Column::const_iterator . . . . .	81
Catch::Matchers::ContainsElementMatcher< T, Equality > . . . . .	
Matcher for checking that an element in range is equal to specific element . . . . .	82
Catch::Matchers::ContainsMatcher< T, AllocComp, AllocMatch > . . . . .	84
Catch::Matchers::ContainsMatcherMatcher< Matcher > . . . . .	
Meta-matcher for checking that an element in a range matches a specific matcher . . . . .	85
Catch::Counts . . . . .	86
Catch::CumulativeReporterBase . . . . .	87
DataBase< TSeq > . . . . .	
Statistical data about the process . . . . .	90
Catch::Decomposer . . . . .	92
Catch::Matchers::EndsWithMatcher . . . . .	93
Entity< TSeq > . . . . .	94
Catch::Detail::EnumInfo . . . . .	94
Catch::Detail::EnumValuesRegistry . . . . .	95
Catch::Benchmark::Environment< Clock > . . . . .	96
Catch::Benchmark::EnvironmentEstimate< Duration > . . . . .	97
Catch::Matchers::EqualsMatcher< T, AllocComp, AllocMatch > . . . . .	98
Catch::ErrnoGuard . . . . .	99
Catch::Benchmark::Estimate< Duration > . . . . .	99
Catch::EventListenerBase . . . . .	100
Catch::EventListenerFactory . . . . .	102
Catch::Matchers::ExceptionMessageMatcher . . . . .	103
Catch::ExceptionTranslatorRegistrar . . . . .	104
Catch::ExceptionTranslatorRegistry . . . . .	104
Catch::Benchmark::ExecutionPlan< Duration > . . . . .	105

Catch::Clara::ExeName	106
Catch::ExprLhs< LhsT >	107
Catch::Clara::Detail::fake_arg	107
Catch::FatalConditionHandler	108
Catch::FatalConditionHandlerGuard	
Simple RAII guard for (dis)engaging the FatalConditionHandler	108
Catch::Generators::FilterGenerator< T, Predicate >	109
Catch::TestSpec::FilterMatch	110
Catch::Generators::FixedValuesGenerator< T >	111
Catch::GeneratorException	112
Catch::Generators::Generators< T >	113
Catch::Generators::GeneratorUntypedBase	115
Catch::Generators::GeneratorWrapper< T >	116
Catch::Detail::has_description< T, typename >	117
Catch::Detail::has_description< T, void_t< decltype(T::getDescription())> >	118
Catch::Matchers::HasSizeMatcher	119
Catch::Clara::Help	120
Catch::Clara::Detail::HelpColumns	121
Catch::IConfig	122
Catch::IContext	123
Catch::IEventListener	124
Catch::IExceptionTranslator	127
Catch::IExceptionTranslatorRegistry	127
Catch::Generators::IGenerator< T >	128
Catch::IGeneratorTracker	129
Catch::IMutableContext	129
Catch::IMutableEnumValuesRegistry	130
Catch::IMutableRegistryHub	131
Catch::IRegistryHub	131
Catch::IReporterFactory	131
Catch::IReporterRegistry	132
Catch::IResultCapture	133
Catch::is_callable< T >	134
Catch::is_callable< Fun(Args...)>	134
Catch::is_callable_tester	135
Catch::is_range< T >	135
Catch::Detail::is_range_impl< T, typename >	136
Catch::Detail::is_range_impl< T, void_t< decltype(begin(std::declval< T >()))> >	137
Catch::Benchmark::Detail::is_related< T, U >	138
Catch::Clara::Detail::is_unary_function< F, typename >	139
Catch::Clara::Detail::is_unary_function< F, Catch::Detail::void_t< decltype(std::declval< F >()(fake_arg()))> >	140
Catch::Matchers::IsEmptyMatcher	141
Catch::ISingleton	142
Catch::Detail::IsStreamInsertable< T >	143
Catch::IStream	143
Catch::ITagAliasRegistry	144
Catch::TextFlow::Columns::iterator	144
Catch::Generators::IteratorGenerator< T >	145
Catch::ITestCaseRegistry	146
Catch::ITestInvoker	147
Catch::TestCaseTracking::ITracker	147
Catch::ITransientExpression	149
Catch::JUnitReporter	150
Catch::Clara::Detail::LambdaInvoker< ReturnType >	151
Catch::Clara::Detail::LambdaInvoker< void >	152
Catch::LazyExpression	152
Catch::LeakDetector	152

LFMCMC< TData >	
Likelihood-Free Markov Chain Monte Carlo	152
Catch::lineOfChars	154
Catch::ListenerDescription	154
Catch::ListenerRegistrar< T >	155
Catch::Detail::make_void<... >	155
Catch::Generators::MapGenerator< T, U, Func >	155
Catch::Matchers::Detail::MatchAllOf< ArgT >	157
Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs >	158
Catch::Matchers::Detail::MatchAnyOf< ArgT >	160
Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs >	161
Catch::Matchers::MatcherBase< T >	163
Catch::Matchers::MatcherGenericBase	164
Catch::Matchers::MatcherUntypedBase	166
Catch::MatchExpr< ArgT, MatcherT >	167
Catch::Matchers::Detail::MatchNotOf< ArgT >	168
Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT >	170
Catch::MessageBuilder	171
Catch::MessageInfo	172
Catch::MessageStream	173
Model< TSeq >	
Core class of epiworld	174
Catch::MultiReporter	182
Catch::TestCaseTracking::NameAndLocation	185
Catch::NameAndTags	186
Catch::CumulativeReporterBase::Node< T, ChildNodeT >	186
Catch::Detail::NonCopyable	
Deriving classes become noncopyable and nonmovable	187
Catch::Matchers::NoneMatchMatcher< Matcher >	188
Catch::Benchmark::now< Clock >	189
Catch::Benchmark::Detail::ObjectStorage< T, Destruct >	189
Catch::Clara::Opt	190
Catch::Optional< T >	192
Catch::Benchmark::OutlierClassification	192
Catch::Clara::Parser	193
Catch::Clara::Detail::ParserBase	194
Catch::Clara::Detail::ParserRefImpl< DerivedT >	195
Catch::Clara::Detail::ParseState	196
PersonTools< TSeq >	196
Catch::pluralise	197
Catch::Matchers::PredicateMatcher< T, Predicate >	197
Catch::ProcessedReporterSpec	198
Progress	
A simple progress bar	199
Queue< TSeq >	
Controls which agents are verified at each step	199
RandGraph	200
Catch::Generators::RandomFloatingGenerator< Float >	200
Catch::Generators::RandomIntegerGenerator< Integer >	202
Catch::Generators::RangeGenerator< T >	203
Catch::ratio_string< Ratio >	205
Catch::ratio_string< std::atto >	205
Catch::ratio_string< std::femto >	205
Catch::ratio_string< std::micro >	205
Catch::ratio_string< std::milli >	206
Catch::ratio_string< std::nano >	206
Catch::ratio_string< std::pico >	206
Catch::RedirectedStdErr	206

Catch::RedirectedStdOut	207
Catch::RedirectedStream	207
Catch::RedirectedStreams	207
Catch::Matchers::RegexMatcher	208
Catch::RegistrarForTagAliases	209
Catch::Benchmark::Detail::repeater< Fun >	209
Catch::Generators::RepeatGenerator< T >	210
Catch::ReporterBase	211
Catch::ReporterConfig	214
Catch::ReporterDescription	214
Catch::ReporterFactory< T >	215
Catch::ReporterPreferences	215
Catch::ReporterRegistrar< T >	216
Catch::ReporterRegistry	217
Catch::ReporterSpec	218
Catch::Clara::Detail::ResultBase	218
Catch::ResultDisposition	219
Catch::Clara::Detail::ResultValueBase< T >	219
Catch::Clara::Detail::ResultValueBase< void >	221
Catch::ResultWas	222
Catch::ReusableStringStream	222
Catch::RunContext	223
Catch::Benchmark::SampleAnalysis< Duration >	225
Catch::XmlWriter::ScopedElement	225
Catch::ScopedMessage	226
Catch::Section	227
Catch::SectionEndInfo	228
Catch::SectionInfo	228
Catch::CumulativeReporterBase::SectionNode	229
Catch::SectionStats	230
Catch::TestCaseTracking::SectionTracker	231
Catch::Session	232
Catch::SimplePcg32	233
Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT >	234
Catch::Generators::SingleValueGenerator< T >	235
Catch::Matchers::SizeMatchesMatcher< Matcher >	237
Catch::SonarQubeReporter	238
Catch::SourceLineInfo	239
Catch::Matchers::StartsWithMatcher	240
Catch::StartupExceptionRegistry	241
Catch::StreamEndStop	241
Catch::StreamingReporterBase	241
Catch::Matchers::StringContainsMatcher	244
Catch::Matchers::StringEqualsMatcher	245
Catch::StringMaker< T, typename >	246
Catch::StringMaker< bool >	246
Catch::StringMaker< Catch::Approx >	246
Catch::StringMaker< char * >	246
Catch::StringMaker< char >	247
Catch::StringMaker< char const * >	247
Catch::StringMaker< char[SZ]>	247
Catch::StringMaker< double >	247
Catch::StringMaker< float >	248
Catch::StringMaker< int >	248
Catch::StringMaker< long >	248
Catch::StringMaker< long long >	248
Catch::StringMaker< R C::* >	249

Catch::StringMaker< R, std::enable_if_t< is_range< R >::value &&!::Catch::Detail::IsStreamInsertable< R >::value > >	249
Catch::StringMaker< signed char >	249
Catch::StringMaker< signed char[SZ]>	249
Catch::StringMaker< std::chrono::duration< Value, Ratio > >	250
Catch::StringMaker< std::chrono::duration< Value, std::ratio< 1 > > >	250
Catch::StringMaker< std::chrono::duration< Value, std::ratio< 3600 > > >	250
Catch::StringMaker< std::chrono::duration< Value, std::ratio< 60 > > >	250
Catch::StringMaker< std::chrono::time_point< Clock, Duration > >	251
Catch::StringMaker< std::chrono::time_point< std::chrono::system_clock, Duration > >	251
Catch::StringMaker< std::nullptr_t >	251
Catch::StringMaker< std::string >	251
Catch::StringMaker< std::wstring >	252
Catch::StringMaker< T * >	252
Catch::StringMaker< T[SZ]>	252
Catch::StringMaker< unsigned char >	252
Catch::StringMaker< unsigned char[SZ]>	253
Catch::StringMaker< unsigned int >	253
Catch::StringMaker< unsigned long >	253
Catch::StringMaker< unsigned long long >	253
Catch::StringMaker< wchar_t * >	254
Catch::StringMaker< wchar_t const * >	254
Catch::Matchers::StringMatcherBase	254
Catch::StringRef	255
Catch::Tag	256
Catch::TagAlias	257
Catch::TagAliasRegistry	258
Catch::TagInfo	258
Catch::Generators::TakeGenerator< T >	259
Catch::TAPReporter	261
Catch::TeamCityReporter	263
Catch::TestCaseHandle	264
Catch::TestCaseInfo	265
Catch::TestCaseInfoHasher	266
Catch::TestCaseStats	267
Catch::TestFailureException	
Used to signal that an assertion macro failed	267
Catch::TestInvokerAsFunction	268
Catch::TestInvokerAsMethod< C >	269
Catch::TestRegistry	270
Catch::TestRunInfo	271
Catch::TestRunStats	271
Catch::TestSpec	272
Catch::TestSpecParser	273
Catch::Timer	273
Catch::Benchmark::Timing< Duration, Result >	273
Catch::Clara::Detail::Token	273
Catch::Clara::Detail::TokenStream	274
Tool< TSeq >	
Tools for defending the agent against the virus	274
Tools< TSeq >	
Set of tools (useful for building iterators)	275
Tools_const< TSeq >	
Set of Tools (const) (useful for iterators)	276
Catch::Totals	277
Catch::TestCaseTracking::TrackerBase	278
Catch::TestCaseTracking::TrackerContext	279
Catch::true_given< typename >	280



Catch::Benchmark::Detail::CompleteType< void >::type	280
Catch::UnaryExpr< LhsT >	281
Catch::Clara::Detail::UnaryLambdaTraits< L >	281
Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::*)(Args...) const >	282
Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::*)(ArgT) const >	282
Catch::Detail::unique_ptr< T >	282
Catch::Matchers::UnorderedEqualsMatcher< T, AllocComp, AllocMatch >	283
UserData< TSeq >	
Personalized data by the user	284
vecHasher< T >	
Vector hasher	286
Catch::Matchers::VectorContainsElementMatcher< T, Alloc >	287
Catch::Version	288
Virus< TSeq >	
Virus	289
Viruses< TSeq >	
Set of viruses (useful for building iterators)	291
Viruses_const< TSeq >	
Set of Viruses (const) (useful for iterators)	291
Catch::WaitForKeypress	292
Catch::WarnAbout	292
Catch::WildcardPattern	293
Catch::Matchers::WithinAbsMatcher	293
Catch::Matchers::WithinRelMatcher	295
Catch::Matchers::WithinUlpMatcher	296
Catch::XmlEncode	297
Catch::XmlReporter	298
Catch::XmlWriter	301



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all documented files with brief descriptions:

include/catch2/catch_amalgamated.hpp	303
include/epiworld/adjlist-bones.hpp	??
include/epiworld/adjlist-meat.hpp	??
include/epiworld/agent-bones.hpp	??
include/epiworld/agent-meat-status.hpp	??
include/epiworld/agent-meat.hpp	??
include/epiworld/config.hpp	??
include/epiworld/database-bones.hpp	??
include/epiworld/database-meat.hpp	??
include/epiworld/entity-bones.hpp	??
include/epiworld/epiworld-macros.hpp	??
include/epiworld/epiworld.hpp	??
include/epiworld/misc.hpp	??
include/epiworld/model-bones.hpp	??
include/epiworld/model-meat-print.hpp	??
include/epiworld/model-meat.hpp	??
include/epiworld/progress.hpp	??
include/epiworld/queue-bones.hpp	??
include/epiworld/randgraph.hpp	??
include/epiworld/random_graph.hpp	??
include/epiworld/seq_processing.hpp	??
include/epiworld/tool-bones.hpp	??
include/epiworld/tool-meat.hpp	??
include/epiworld/tools-bones.hpp	??
include/epiworld/userdata-bones.hpp	??
include/epiworld/userdata-meat.hpp	??
include/epiworld/virus-bones.hpp	??
include/epiworld/virus-meat.hpp	??
include/epiworld/viruses-bones.hpp	??
include/epiworld/math/lfmcmc.hpp	??
include/epiworld/math/lfmcmc/lfmcmc-bones.hpp	??
include/epiworld/math/lfmcmc/lfmcmc-meat.hpp	??
include/epiworld/models/immune_system.hpp	??
include/epiworld/models/seirconnected.hpp	??
include/epiworld/models/sir.hpp	??
include/epiworld/models/sirconnected.hpp	??
include/epiworld/models/surveillance.hpp	??



## Chapter 5

# Class Documentation

### 5.1 Catch::Clara::accept\_many\_t Struct Reference

The documentation for this struct was generated from the following file:

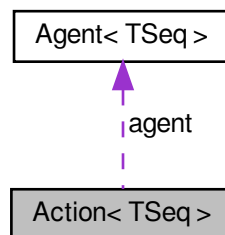
- [include/catch2/catch\\_amalgamated.hpp](#)

### 5.2 Action< TSeq > Struct Template Reference

[Action](#) data for update an agent.

```
#include <config.hpp>
```

Collaboration diagram for Action< TSeq >:



#### Public Member Functions

- [Action](#) ([Agent](#)< TSeq > \*agent\_, VirusPtr< TSeq > virus\_, ToolPtr< TSeq > tool\_, epiworld\_fast\_int new←\_status\_, epiworld\_fast\_int queue\_, ActionFun< TSeq > call\_)
- Construct a new [Action](#) object.*

## Public Attributes

- [Agent](#)< TSeq > \* **agent**
- VirusPtr< TSeq > **virus**
- ToolPtr< TSeq > **tool**
- epiworld\_fast\_int **new\_status**
- epiworld\_fast\_int **queue**
- ActionFun< TSeq > **call**

### 5.2.1 Detailed Description

```
template<typename TSeq>
struct Action< TSeq >
```

[Action](#) data for update an agent.

Template Parameters

<i>TSeq</i>	
-------------	--

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 Action()

```
template<typename TSeq >
Action< TSeq >::Action (
    Agent< TSeq > * agent_,
    VirusPtr< TSeq > virus_,
    ToolPtr< TSeq > tool_,
    epiworld_fast_int new_status_,
    epiworld_fast_int queue_,
    ActionFun< TSeq > call_ ) [inline]
```

Construct a new [Action](#) object.

All the parameters are rather optional.

Parameters

<i>agent_</i>	<a href="#">Agent</a> over who the action will happen
<i>virus_</i>	<a href="#">Virus</a> to add
<i>tool_</i>	<a href="#">Tool</a> to add
<i>virus_idx</i>	Index of virus to be removed (if needed)
<i>tool_idx</i>	Index of tool to be removed (if needed)
<i>new_↔ status_</i>	Next status
<i>queue_</i>	Effect on the queue
<i>call_</i>	The action call (if needed)

The documentation for this struct was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/config.hpp

## 5.3 AdjList Class Reference

### Public Member Functions

- [AdjList](#) (const std::vector< unsigned int > &source, const std::vector< unsigned int > &target, int size, bool directed)  
*Construct a new Adj List object.*
- void [read\\_edgelist](#) (std::string fn, int size, int skip=0, bool directed=true)  
*Read an edgelist.*
- std::map< unsigned int, unsigned int > **operator()** (unsigned int i) const
- void **print** (unsigned int limit=20u) const
- size\_t [vcount](#) () const  
*Number of vertices/nodes in the network.*
- size\_t [ecount](#) () const  
*Number of edges/arcs/ties in the network.*
- std::vector< std::map< unsigned int, unsigned int > > & **get\_dat** ()
- bool [is\\_directed](#) () const  
*true if the network is directed.*

### 5.3.1 Constructor & Destructor Documentation

#### 5.3.1.1 AdjList()

```
AdjList::AdjList (
    const std::vector< unsigned int > & source,
    const std::vector< unsigned int > & target,
    int size,
    bool directed ) [inline]
```

Construct a new Adj List object.

Ids in the network are assume to range from 0 to `size - 1`.

#### Parameters

<i>source</i>	Unsigned int vector with the source
<i>target</i>	Unsigned int vector with the target
<i>size</i>	Number of vertices in the network.
<i>directed</i>	Bool true if the network is directed

## 5.3.2 Member Function Documentation

### 5.3.2.1 read\_edgelist()

```
void AdjList::read_edgelist (
    std::string fn,
    int size,
    int skip = 0,
    bool directed = true ) [inline]
```

Read an edgelist.

Ids in the network are assume to range from 0 to `size - 1`.

#### Parameters

<i>fn</i>	Path to the file
<i>skip</i>	Number of lines to skip (e.g., 1 if there's a header)
<i>directed</i>	<code>true</code> if the network is directed
<i>size</i>	Number of vertices in the network.

The documentation for this class was generated from the following files:

- `include/epiworld/adjlist-bones.hpp`
- `include/epiworld/adjlist-meat.hpp`

## 5.4 Agent< TSeq > Class Template Reference

[Agent](#) (agents)

```
#include <agent-bones.hpp>
```

### Public Member Functions

- **Agent** (const [Agent](#)< TSeq > &p)
- int [get\\_id](#) () const  
*Id of the individual.*
- std::mt19937 \* **get\_rand\_engine** ()
- [Model](#)< TSeq > \* **get\_model** ()
- VirusPtr< TSeq > & **get\_virus** (int i)
- [Viruses](#)< TSeq > **get\_viruses** ()
- const [Viruses\\_const](#)< TSeq > **get\_viruses** () const
- size\_t **get\_n\_viruses** () const noexcept
- ToolPtr< TSeq > & **get\_tool** (int i)
- [Tools](#)< TSeq > **get\_tools** ()
- const [Tools\\_const](#)< TSeq > **get\_tools** () const



- `size_t get_n_tools ()` const noexcept
- `void mutate_variant ()`
- `void add_neighbor (Agent< TSeq > *p, bool check_source=true, bool check_target=true)`
- `std::vector< Agent< TSeq > * > & get_neighbors ()`
- `void change_status (epiworld_fast_uint new_status, epiworld_fast_int queue=0)`
- `const epiworld_fast_uint & get_status ()` const
- `void reset ()`
- `bool has_tool (unsigned int t)` const
- `bool has_tool (std::string name)` const
- `bool has_virus (unsigned int t)` const
- `bool has_virus (std::string name)` const

### Add/Remove Virus/Tool

*Any of these is ultimately reflected at the end of the iteration.*

#### Parameters

tool	<i>Tool to add</i>
virus	<i>Virus to add</i>
status_new	<i>Status after the change</i>
queue	

- `void add_tool (ToolPtr< TSeq > tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void add_tool (Tool< TSeq > tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void add_virus (VirusPtr< TSeq > virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void add_virus (Virus< TSeq > virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void rm_tool (epiworld_fast_uint tool_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void rm_tool (ToolPtr< TSeq > &tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void rm_virus (epiworld_fast_uint virus_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void rm_virus (VirusPtr< TSeq > &virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`
- `void rm_agent_by_virus (epiworld_fast_uint virus_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`  
*Agent removed by virus.*
- `void rm_agent_by_virus (VirusPtr< TSeq > &virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)`  
*Agent removed by virus.*

### Get the rates (multipliers) for the agent

#### Parameters

v	<i>A pointer to a virus.</i>
---	------------------------------

#### Returns

*epiworld\_double*

- `epiworld_double get_susceptibility_reduction (VirusPtr< TSeq > v)`
- `epiworld_double get_transmission_reduction (VirusPtr< TSeq > v)`
- `epiworld_double get_recovery_enhancer (VirusPtr< TSeq > v)`
- `epiworld_double get_death_reduction (VirusPtr< TSeq > v)`

## Friends

- class **Model**< TSeq >
- class **Virus**< TSeq >
- class **Viruses**< TSeq >
- class **Viruses\_const**< TSeq >
- class **Tool**< TSeq >
- class **Tools**< TSeq >
- class **Queue**< TSeq >
- void **default\_add\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_add\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)

### 5.4.1 Detailed Description

```
template<typename TSeq = int>
class Agent< TSeq >
```

[Agent](#) (agents)

Template Parameters

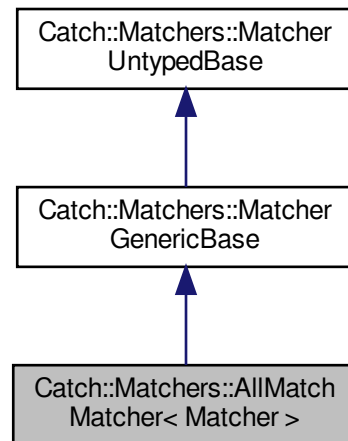
<i>TSeq</i>	Sequence type (should match TSeq across the model)
-------------	--

The documentation for this class was generated from the following file:

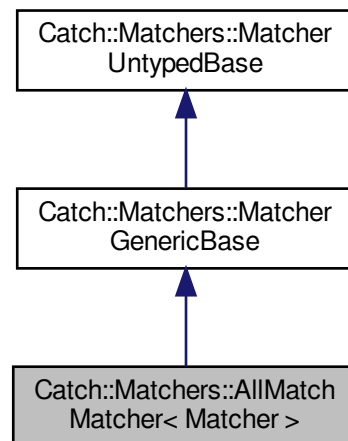
- include/epiworld/agent-bones.hpp

## 5.5 Catch::Matchers::AllMatchMatcher< Matcher > Class Template Reference

Inheritance diagram for Catch::Matchers::AllMatchMatcher< Matcher >:



Collaboration diagram for Catch::Matchers::AllMatchMatcher< Matcher >:



### Public Member Functions

- **AllMatchMatcher** (Matcher matcher)
- `std::string describe ()` const override
- `template<typename RangeLike >`  
`bool match (RangeLike &&rng)` const

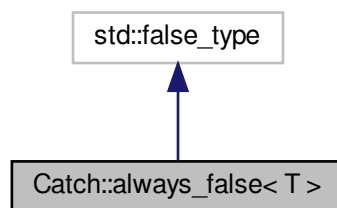
## Additional Inherited Members

The documentation for this class was generated from the following file:

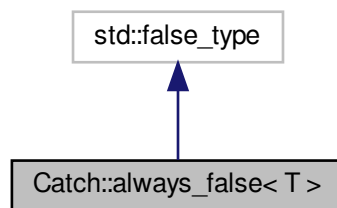
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.6 Catch::always\_false< T > Struct Template Reference

Inheritance diagram for Catch::always\_false< T >:



Collaboration diagram for Catch::always\_false< T >:

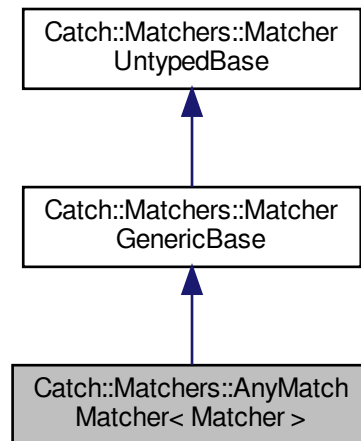


The documentation for this struct was generated from the following file:

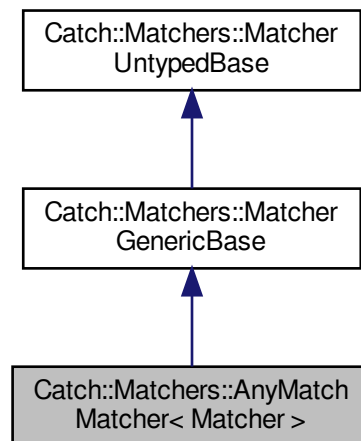
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.7 Catch::Matchers::AnyMatchMatcher< Matcher > Class Template Reference

Inheritance diagram for Catch::Matchers::AnyMatchMatcher< Matcher >:



Collaboration diagram for Catch::Matchers::AnyMatchMatcher< Matcher >:



### Public Member Functions

- **AnyMatchMatcher** (Matcher matcher)
- std::string **describe** () const override
- template<typename RangeLike >  
bool **match** (RangeLike &&rng) const

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.8 Catch::Approx Class Reference

### Public Member Functions

- **Approx** (double value)
- **Approx operator-** () const
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
**Approx operator()** (T const &value) const
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
**Approx** (T const &value)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
**Approx & epsilon** (T const &newEpsilon)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
**Approx & margin** (T const &newMargin)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
**Approx & scale** (T const &newScale)
- std::string **toString** () const

### Static Public Member Functions

- static **Approx custom** ()

### Friends

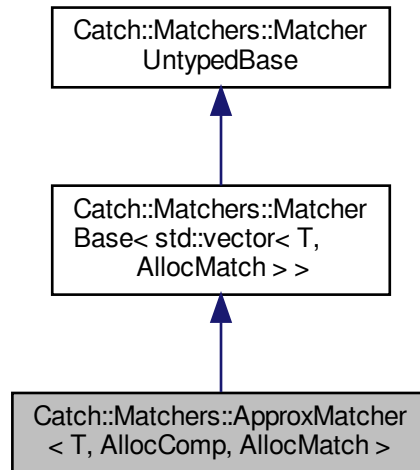
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator==** (const T &lhs, **Approx** const &rhs)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator==** (**Approx** const &lhs, const T &rhs)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator!=** (T const &lhs, **Approx** const &rhs)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator!=** (**Approx** const &lhs, T const &rhs)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator<=** (T const &lhs, **Approx** const &rhs)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator<=** (**Approx** const &lhs, T const &rhs)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator>=** (T const &lhs, **Approx** const &rhs)
- template<typename T , typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
bool **operator>=** (**Approx** const &lhs, T const &rhs)

The documentation for this class was generated from the following file:

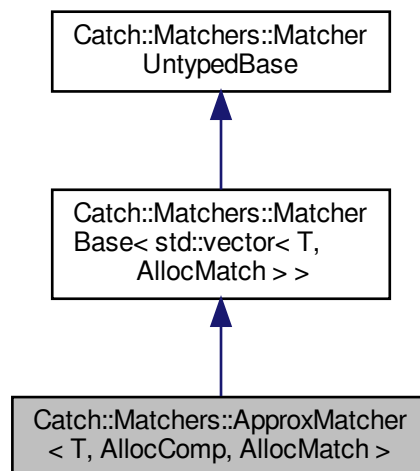
- include/catch2/catch\_amalgamated.hpp

## 5.9 Catch::Matchers::ApproxMatcher< T, AllocComp, AllocMatch > Class Template Reference

Inheritance diagram for Catch::Matchers::ApproxMatcher< T, AllocComp, AllocMatch >:



Collaboration diagram for Catch::Matchers::ApproxMatcher< T, AllocComp, AllocMatch >:



### Public Member Functions

- **ApproxMatcher** (std::vector< T, AllocComp > const &comparator)

- bool **match** (std::vector< T, AllocMatch > const &v) const override
- std::string **describe** () const override
- template<typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
[ApproxMatcher](#) & **epsilon** (T const &newEpsilon)
- template<typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
[ApproxMatcher](#) & **margin** (T const &newMargin)
- template<typename = std::enable\_if\_t<std::is\_constructible<double, T>::value>>  
[ApproxMatcher](#) & **scale** (T const &newScale)

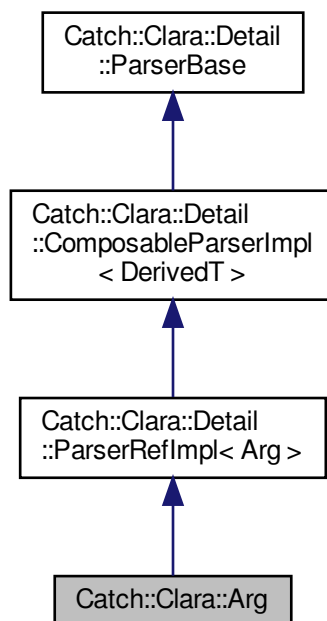
## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

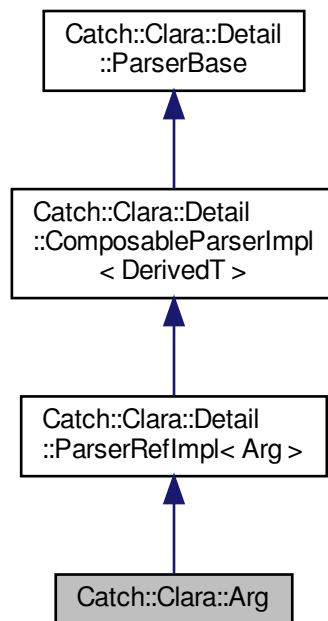
## 5.10 Catch::Clara::Arg Class Reference

Inheritance diagram for Catch::Clara::Arg:





Collaboration diagram for Catch::Clara::Arg:



## Public Member Functions

- [Detail::InternalParseResult](#) **parse** (std::string const &, [Detail::TokenStream](#) const &tokens) const override

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.11 Catch::Clara::Args Class Reference

### Public Member Functions

- **Args** (int argc, char const \*const \*argv)
- **Args** (std::initializer\_list< std::string > args)
- std::string const & **exeName** () const

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.12 Catch::Generators::as< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.13 Catch::AssertionHandler Class Reference

### Public Member Functions

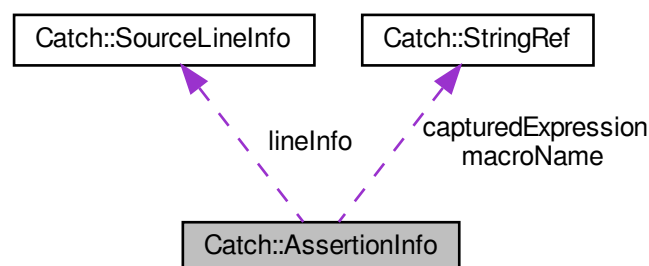
- **AssertionHandler** ([StringRef](#) macroName, [SourceLineInfo](#) const &lineInfo, [StringRef](#) capturedExpression, ResultDisposition::Flags resultDisposition)
- `template<typename T >`  
void **handleExpr** ([ExprLhs](#)< T > const &expr)
- void **handleExpr** ([ITransientExpression](#) const &expr)
- void **handleMessage** (ResultWas::OfType resultType, [StringRef](#) message)
- void **handleExceptionThrownAsExpected** ()
- void **handleUnexpectedExceptionNotThrown** ()
- void **handleExceptionNotThrownAsExpected** ()
- void **handleThrowingCallSkipped** ()
- void **handleUnexpectedInflightException** ()
- void **complete** ()
- void **setCompleted** ()
- `auto` **allowThrows** () const -> bool

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.14 Catch::AssertionInfo Struct Reference

Collaboration diagram for Catch::AssertionInfo:



## Public Attributes

- [StringRef](#) **macroName**
- [SourceLineInfo](#) **lineInfo**
- [StringRef](#) **capturedExpression**
- [ResultDisposition::Flags](#) **resultDisposition**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.15 Catch::Detail::AssertionOrBenchmarkResult Class Reference

Represents either an assertion or a benchmark result to be handled by cumulative reporter later.

```
#include <catch_amalgamated.hpp>
```

## Public Member Functions

- **AssertionOrBenchmarkResult** ([AssertionStats](#) const &assertion)
- **AssertionOrBenchmarkResult** ([BenchmarkStats](#)<> const &benchmark)
- bool **isAssertion** () const
- bool **isBenchmark** () const
- [AssertionStats](#) const & **asAssertion** () const
- [BenchmarkStats](#) const & **asBenchmark** () const

### 5.15.1 Detailed Description

Represents either an assertion or a benchmark result to be handled by cumulative reporter later.

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.16 Catch::AssertionReaction Struct Reference

## Public Attributes

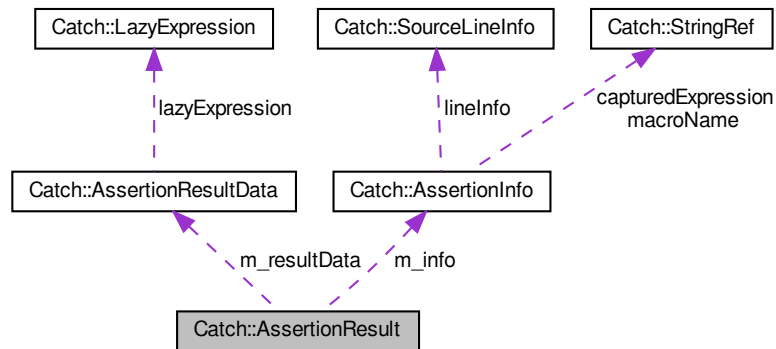
- bool **shouldDebugBreak** = false
- bool **shouldThrow** = false

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.17 Catch::AssertionResult Class Reference

Collaboration diagram for Catch::AssertionResult:



### Public Member Functions

- **AssertionResult** ([AssertionInfo](#) const &info, [AssertionResultData](#) const &data)
- bool **isOk** () const
- bool **succeeded** () const
- ResultWas::OfType **getResultType** () const
- bool **hasExpression** () const
- bool **hasMessage** () const
- std::string **getExpression** () const
- std::string **getExpressionInMacro** () const
- bool **hasExpandedExpression** () const
- std::string **getExpandedExpression** () const
- [StringRef](#) **getMessage** () const
- [SourceLineInfo](#) **getSourceInfo** () const
- [StringRef](#) **getTestMacroName** () const

### Public Attributes

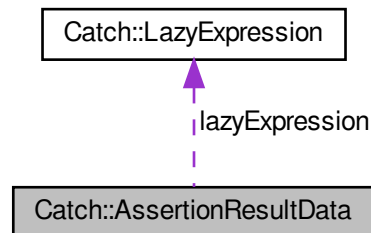
- [AssertionInfo](#) **m\_info**
- [AssertionResultData](#) **m\_resultData**

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.18 Catch::AssertionResultData Struct Reference

Collaboration diagram for Catch::AssertionResultData:



### Public Member Functions

- **AssertionResultData** (ResultWas::OfType \_resultType, [LazyExpression](#) const &\_lazyExpression)
- std::string **reconstructExpression** () const

### Public Attributes

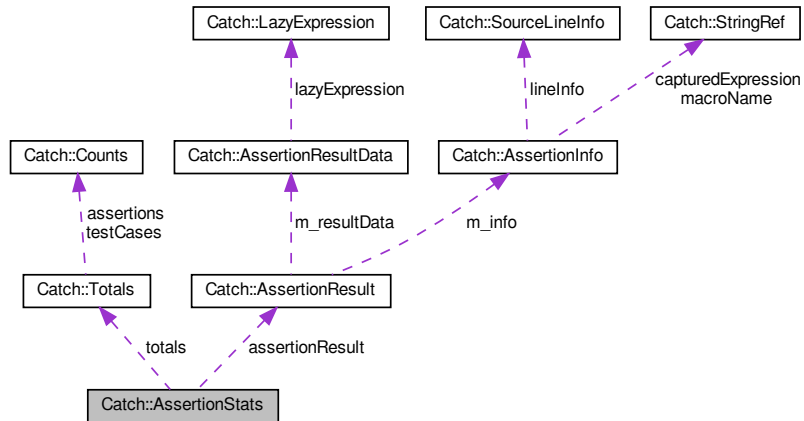
- std::string **message**
- std::string **reconstructedExpression**
- [LazyExpression](#) **lazyExpression**
- ResultWas::OfType **resultType**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.19 Catch::AssertionStats Struct Reference

Collaboration diagram for Catch::AssertionStats:



### Public Member Functions

- **AssertionStats** ([AssertionResult](#) const &\_assertionResult, std::vector< [MessageInfo](#) > const &\_info← Messages, [Totals](#) const &\_totals)
- **AssertionStats** ([AssertionStats](#) const &)=default
- **AssertionStats** ([AssertionStats](#) &&)=default
- [AssertionStats](#) & **operator=** ([AssertionStats](#) const &)=delete
- [AssertionStats](#) & **operator=** ([AssertionStats](#) &&)=delete

### Public Attributes

- [AssertionResult](#) **assertionResult**
- std::vector< [MessageInfo](#) > **infoMessages**
- [Totals](#) **totals**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



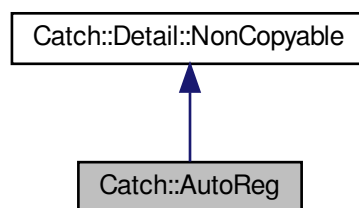
## Additional Inherited Members

The documentation for this class was generated from the following file:

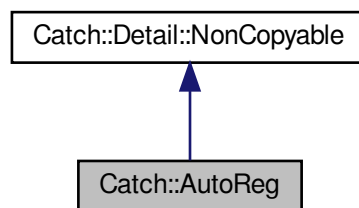
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.21 Catch::AutoReg Struct Reference

Inheritance diagram for Catch::AutoReg:



Collaboration diagram for Catch::AutoReg:



## Public Member Functions

- **AutoReg** ([Detail::unique\\_ptr](#)< [ITestInvoker](#) > invoker, [SourceLineInfo](#) const &lineInfo, [StringRef](#) classOrMethod, [NameAndTags](#) const &nameAndTags) noexcept

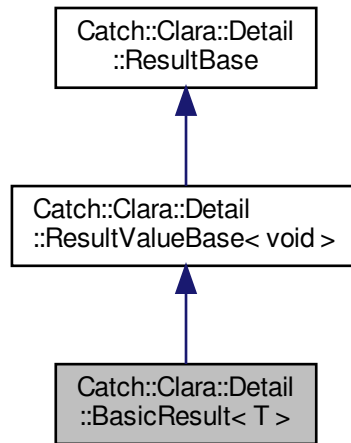
The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

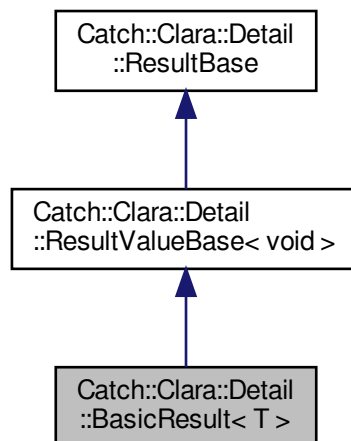


## 5.22 Catch::Clara::Detail::BasicResult< T > Class Template Reference

Inheritance diagram for Catch::Clara::Detail::BasicResult< T >:



Collaboration diagram for Catch::Clara::Detail::BasicResult< T >:



### Public Member Functions

- `template<typename U >`  
**BasicResult** ([BasicResult](#)< U > const &other)
- **operator bool** () const
- `auto type` () const -> [ResultType](#)
- `auto errorMessage` () const -> `std::string` const &

## Static Public Member Functions

- `template<typename U >`  
`static auto ok (U const &value) -> BasicResult`
- `static auto ok () -> BasicResult`
- `static auto logicError (std::string &&message) -> BasicResult`
- `static auto runtimeError (std::string &&message) -> BasicResult`

## Protected Member Functions

- `void enforceOk () const override`
- `BasicResult (ResultType type, std::string &&message)`

## Protected Attributes

- `std::string m_errorMessage`
- `ResultType m_type`

The documentation for this class was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.23 Catch::Benchmark::Benchmark Struct Reference

### Public Member Functions

- `Benchmark (std::string &&benchmarkName)`
- `template<class FUN >`  
`Benchmark (std::string &&benchmarkName, FUN &&func)`
- `template<typename Clock >`  
`ExecutionPlan< FloatDuration< Clock > > prepare (const IConfig &cfg, Environment< FloatDuration< Clock >> env) const`
- `template<typename Clock = default_clock>`  
`void run ()`
- `template<typename Fun , std::enable_if_t<!Detail::is_related< Fun, Benchmark >::value, int > = 0>`  
`Benchmark & operator= (Fun func)`
- `operator bool ()`

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.24 Catch::Benchmark::Detail::BenchmarkFunction Struct Reference

```
#include <catch_amalgamated.hpp>
```

## Public Member Functions

- `template<typename Fun , std::enable_if_t<!is_related< Fun, BenchmarkFunction >::value, int > = 0>`  
**BenchmarkFunction** (Fun &&fun)
- **BenchmarkFunction** ([BenchmarkFunction](#) &&that) noexcept
- **BenchmarkFunction** ([BenchmarkFunction](#) const &that)
- [BenchmarkFunction](#) & **operator=** ([BenchmarkFunction](#) &&that) noexcept
- [BenchmarkFunction](#) & **operator=** ([BenchmarkFunction](#) const &that)
- void **operator()** ([Chronometer](#) meter) const

### 5.24.1 Detailed Description

We need to reinvent `std::function` because every piece of code that might add overhead in a measurement context needs to have consistent performance characteristics so that we can account for it in the measurement. Implementations of `std::function` with optimizations that aren't always applicable, like small buffer optimizations, are not uncommon. This is effectively an implementation of `std::function` without any such optimizations; it may be slow, but it is consistently slow.

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.25 Catch::BenchmarkInfo Struct Reference

### Public Attributes

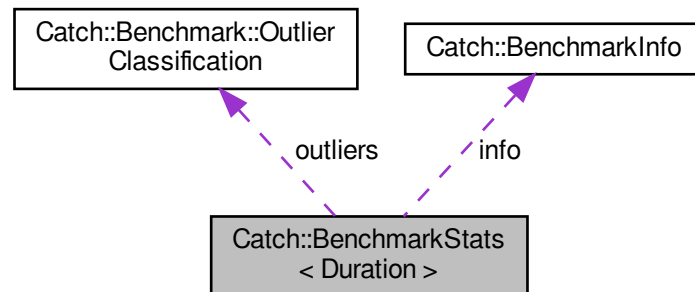
- `std::string` **name**
- `double` **estimatedDuration**
- `int` **iterations**
- `unsigned int` **samples**
- `unsigned int` **resamples**
- `double` **clockResolution**
- `double` **clockCost**

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.26 Catch::BenchmarkStats< Duration > Struct Template Reference

Collaboration diagram for Catch::BenchmarkStats< Duration >:



### Public Member Functions

- `template<typename Duration2 >`  
**operator BenchmarkStats< Duration2 > () const**

### Public Attributes

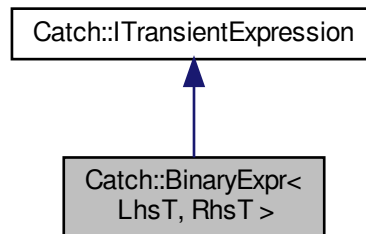
- [BenchmarkInfo](#) **info**
- `std::vector< Duration >` **samples**
- [Benchmark::Estimate< Duration >](#) **mean**
- [Benchmark::Estimate< Duration >](#) **standardDeviation**
- [Benchmark::OutlierClassification](#) **outliers**
- `double` **outlierVariance**

The documentation for this struct was generated from the following file:

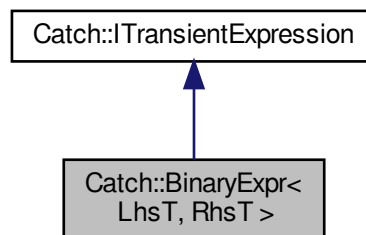
- `include/catch2/catch_amalgamated.hpp`

## 5.27 Catch::BinaryExpr< LhsT, RhsT > Class Template Reference

Inheritance diagram for Catch::BinaryExpr< LhsT, RhsT >:



Collaboration diagram for Catch::BinaryExpr< LhsT, RhsT >:



### Public Member Functions

- **BinaryExpr** (bool comparisonResult, LhsT lhs, [StringRef](#) op, RhsT rhs)
- template<typename T >  
auto **operator&&** (T) const -> [BinaryExpr](#)< LhsT, RhsT const & > const
- template<typename T >  
auto **operator||** (T) const -> [BinaryExpr](#)< LhsT, RhsT const & > const
- template<typename T >  
auto **operator==** (T) const -> [BinaryExpr](#)< LhsT, RhsT const & > const
- template<typename T >  
auto **operator!=** (T) const -> [BinaryExpr](#)< LhsT, RhsT const & > const
- template<typename T >  
auto **operator>** (T) const -> [BinaryExpr](#)< LhsT, RhsT const & > const
- template<typename T >  
auto **operator<** (T) const -> [BinaryExpr](#)< LhsT, RhsT const & > const
- template<typename T >  
auto **operator>=** (T) const -> [BinaryExpr](#)< LhsT, RhsT const & > const

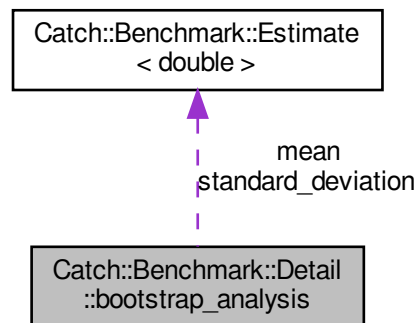
- `template<typename T >`  
`auto operator<= (T) const -> BinaryExpr< LhsT, Rhst const & > const`

The documentation for this class was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.28 `Catch::Benchmark::Detail::bootstrap_analysis` Struct Reference

Collaboration diagram for `Catch::Benchmark::Detail::bootstrap_analysis`:



### Public Attributes

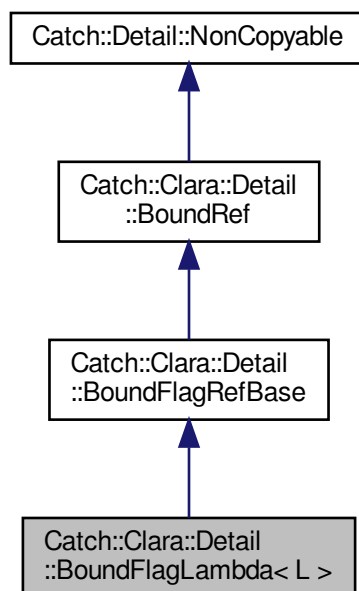
- `Estimate< double > mean`
- `Estimate< double > standard_deviation`
- `double outlier_variance`

The documentation for this struct was generated from the following file:

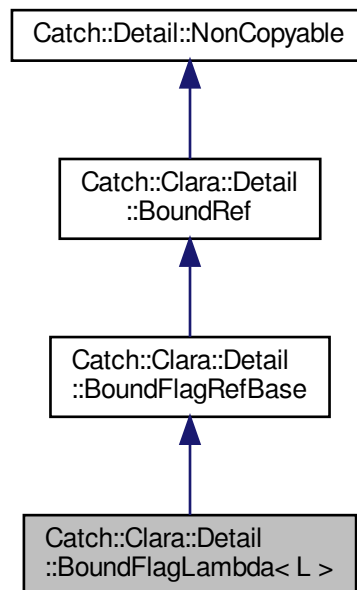
- `include/catch2/catch\_amalgamated.hpp`

## 5.29 Catch::Clara::Detail::BoundFlagLambda< L > Struct Template Reference

Inheritance diagram for Catch::Clara::Detail::BoundFlagLambda< L >:



Collaboration diagram for `Catch::Clara::Detail::BoundFlagLambda< L >`:



## Public Member Functions

- **BoundFlagLambda** (L const &lambda)
- auto **setFlag** (bool flag) -> [ParserResult](#) override

## Public Attributes

- L **m\_lambda**

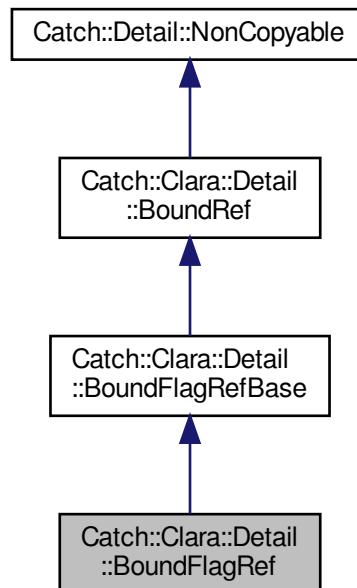
The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

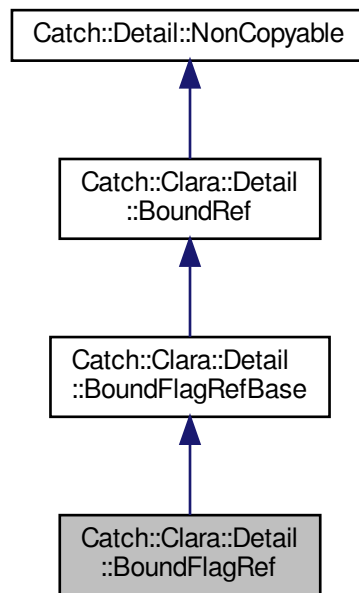


## 5.30 Catch::Clara::Detail::BoundFlagRef Struct Reference

Inheritance diagram for Catch::Clara::Detail::BoundFlagRef:



Collaboration diagram for Catch::Clara::Detail::BoundFlagRef:



## Public Member Functions

- **BoundFlagRef** (bool &ref)
- [ParserResult](#) **setFlag** (bool flag) override

## Public Attributes

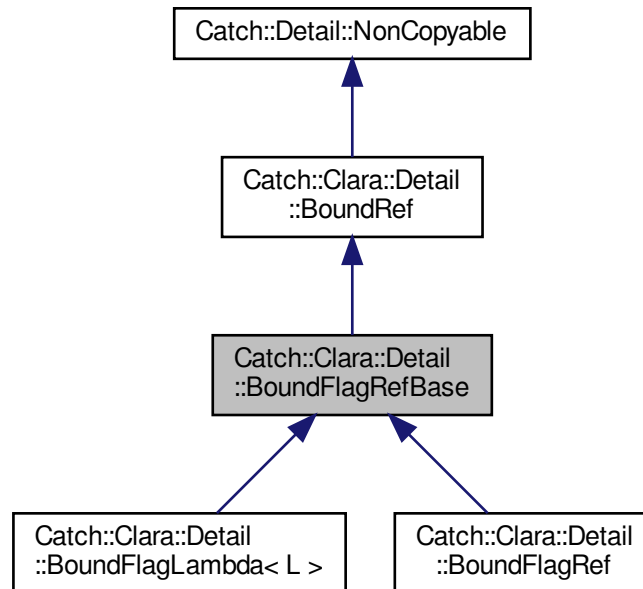
- bool & **m\_ref**

The documentation for this struct was generated from the following file:

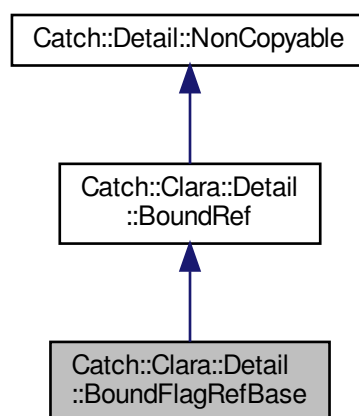
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.31 Catch::Clara::Detail::BoundFlagRefBase Struct Reference

Inheritance diagram for Catch::Clara::Detail::BoundFlagRefBase:



Collaboration diagram for Catch::Clara::Detail::BoundFlagRefBase:



### Public Member Functions

- virtual auto **setFlag** (bool flag) -> [ParserResult=0](#)

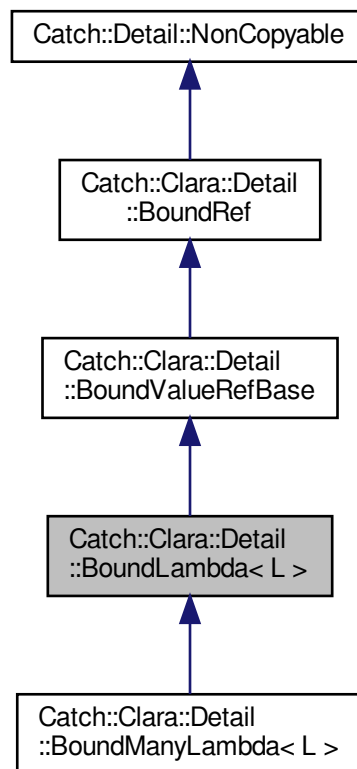
- `bool isFlag ()` const override

The documentation for this struct was generated from the following file:

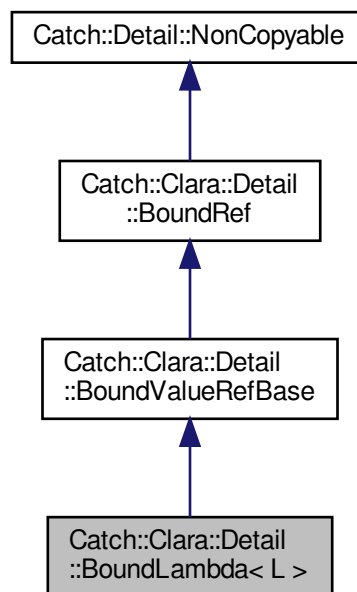
- `include/catch2/catch_amalgamated.hpp`

### 5.32 Catch::Clara::Detail::BoundLambda< L > Struct Template Reference

Inheritance diagram for Catch::Clara::Detail::BoundLambda< L >:



Collaboration diagram for Catch::Clara::Detail::BoundLambda< L >:



### Public Member Functions

- **BoundLambda** (L const &lambda)
- auto **setValue** (std::string const &arg) -> [ParserResult](#) override

### Public Attributes

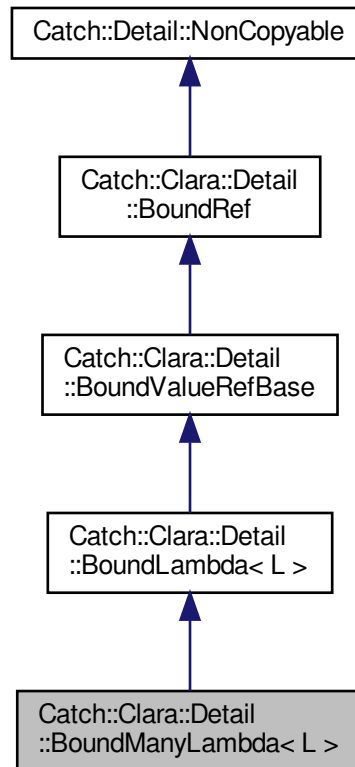
- L **m\_lambda**

The documentation for this struct was generated from the following file:

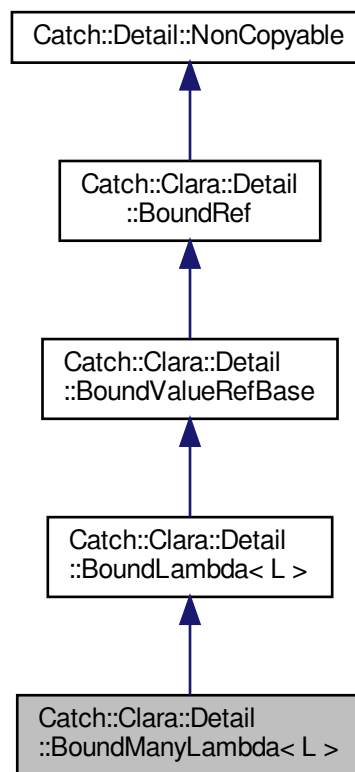
- include/catch2/[catch\\_amalgamated.hpp](#)

### 5.33 Catch::Clara::Detail::BoundManyLambda< L > Struct Template Reference

Inheritance diagram for Catch::Clara::Detail::BoundManyLambda< L >:



Collaboration diagram for Catch::Clara::Detail::BoundManyLambda< L >:



### Public Member Functions

- **BoundManyLambda** (L const &lambda)
- **bool isContainer** () const override

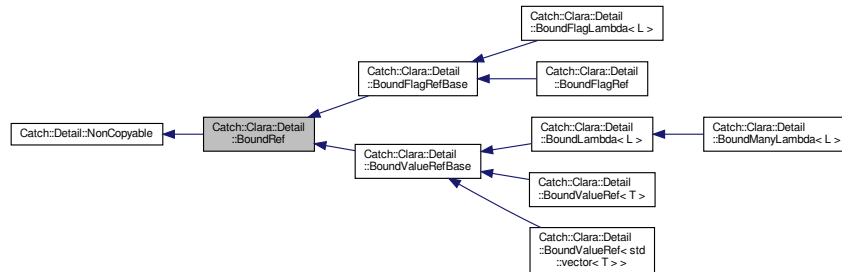
### Additional Inherited Members

The documentation for this struct was generated from the following file:

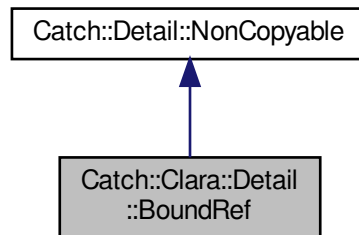
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.34 Catch::Clara::Detail::BoundRef Struct Reference

Inheritance diagram for Catch::Clara::Detail::BoundRef:



Collaboration diagram for Catch::Clara::Detail::BoundRef:



### Public Member Functions

- virtual bool **isContainer** () const
- virtual bool **isFlag** () const

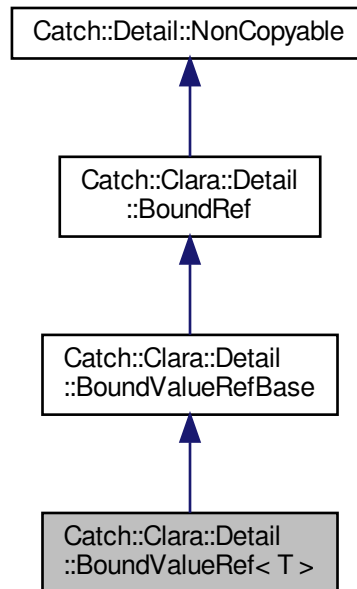
The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

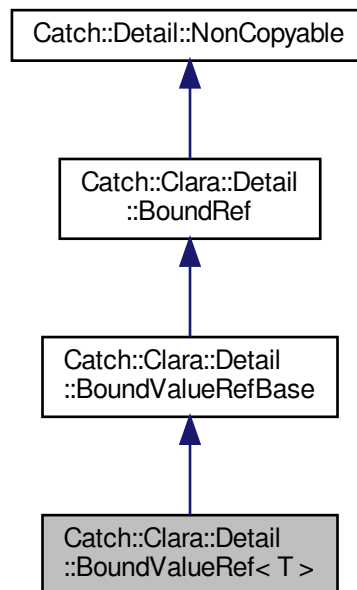


## 5.35 Catch::Clara::Detail::BoundValueRef< T > Struct Template Reference

Inheritance diagram for Catch::Clara::Detail::BoundValueRef< T >:



Collaboration diagram for `Catch::Clara::Detail::BoundValueRef< T >`:



## Public Member Functions

- **BoundValueRef** (T &ref)
- [ParserResult](#) **setValue** (std::string const &arg) override

## Public Attributes

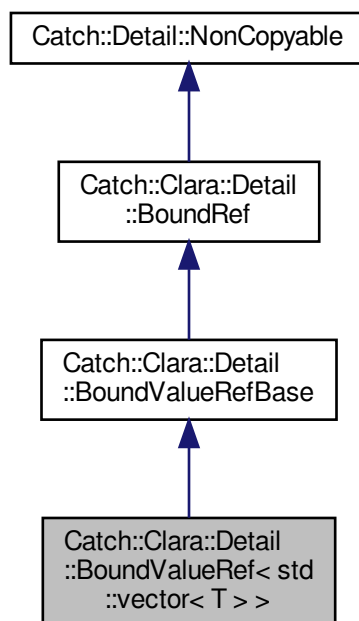
- T & **m\_ref**

The documentation for this struct was generated from the following file:

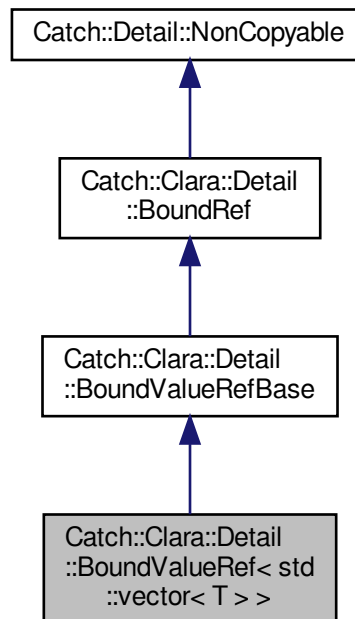
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.36 Catch::Clara::Detail::BoundValueRef< std::vector< T > > Struct Template Reference

Inheritance diagram for Catch::Clara::Detail::BoundValueRef< std::vector< T > >:



Collaboration diagram for `Catch::Clara::Detail::BoundValueRef< std::vector< T > >`:



## Public Member Functions

- **BoundValueRef** (`std::vector< T > &ref`)
- `auto isContainer () const -> bool` override
- `auto setValue (std::string const &arg) -> ParserResult` override

## Public Attributes

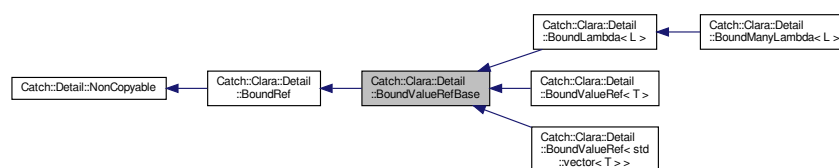
- `std::vector< T > & m_ref`

The documentation for this struct was generated from the following file:

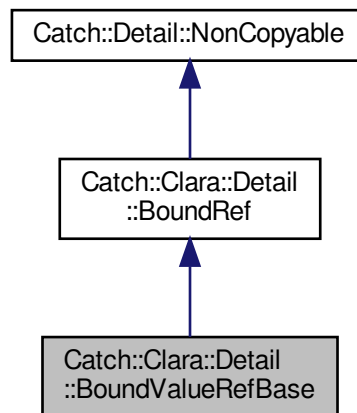
- `include/catch2/catch_amalgamated.hpp`

## 5.37 Catch::Clara::Detail::BoundValueRefBase Struct Reference

Inheritance diagram for `Catch::Clara::Detail::BoundValueRefBase`:



Collaboration diagram for Catch::Clara::Detail::BoundValueRefBase:



## Public Member Functions

- virtual auto **setValue** (std::string const &arg) -> [ParserResult](#)=0

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.38 Catch::Capturer Class Reference

### Public Member Functions

- **Capturer** ([StringRef](#) macroName, [SourceLineInfo](#) const &lineInfo, ResultWas::OfType resultType, [StringRef](#) names)
- **Capturer** ([Capturer](#) const &)=delete
- [Capturer](#) & **operator=** ([Capturer](#) const &)=delete
- void **captureValue** (size\_t index, std::string const &value)
- template<typename T >  
void **captureValues** (size\_t index, T const &value)
- template<typename T, typename... Ts>  
void **captureValues** (size\_t index, T const &value, Ts const &... values)

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.39 Catch::Matchers::CasedString Struct Reference

### Public Member Functions

- **CasedString** (std::string const &str, CaseSensitive caseSensitivity)
- std::string **adjustString** (std::string const &str) const
- [StringRef](#) **caseSensitivitySuffix** () const

### Public Attributes

- CaseSensitive **m\_caseSensitivity**
- std::string **m\_str**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.40 Catch::Detail::CaseInsensitiveEqualTo Struct Reference

Provides case-insensitive `op==` semantics when called.

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- bool **operator()** ([StringRef](#) lhs, [StringRef](#) rhs) const

#### 5.40.1 Detailed Description

Provides case-insensitive `op==` semantics when called.

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.41 Catch::Detail::CaseInsensitiveLess Struct Reference

Provides case-insensitive `op<` semantics when called.

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- bool **operator()** ([StringRef](#) lhs, [StringRef](#) rhs) const

### 5.41.1 Detailed Description

Provides case-insensitive `op<` semantics when called.

The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.42 Catch\_global\_namespace\_dummy Struct Reference

The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.43 Catch::Benchmark::Chronometer Struct Reference

### Public Member Functions

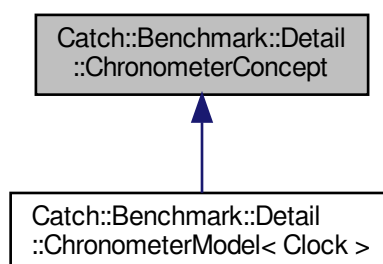
- `template<typename Fun >`  
`void measure (Fun &&fun)`
- `int runs () const`
- `Chronometer (Detail::ChronometerConcept &meter, int repeats_)`

The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.44 Catch::Benchmark::Detail::ChronometerConcept Struct Reference

Inheritance diagram for `Catch::Benchmark::Detail::ChronometerConcept`:



## Public Member Functions

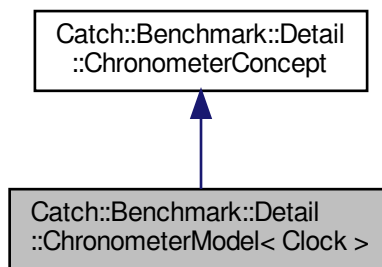
- virtual void **start** ()=0
- virtual void **finish** ()=0
- **ChronometerConcept** ([ChronometerConcept](#) const &)=default
- [ChronometerConcept](#) & **operator=** ([ChronometerConcept](#) const &)=default

The documentation for this struct was generated from the following file:

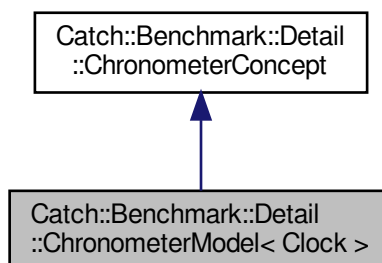
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.45 Catch::Benchmark::Detail::ChronometerModel< Clock > Struct Template Reference

Inheritance diagram for Catch::Benchmark::Detail::ChronometerModel< Clock >:



Collaboration diagram for Catch::Benchmark::Detail::ChronometerModel< Clock >:





## Public Member Functions

- void **start** () override
- void **finish** () override
- ClockDuration< Clock > **elapsed** () const

## Public Attributes

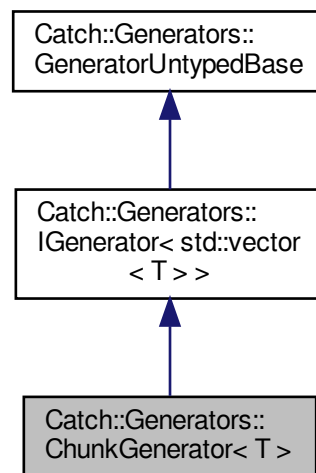
- TimePoint< Clock > **started**
- TimePoint< Clock > **finished**

The documentation for this struct was generated from the following file:

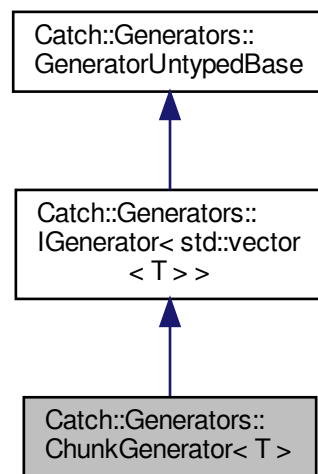
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.46 Catch::Generators::ChunkGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::ChunkGenerator< T >:



Collaboration diagram for `Catch::Generators::ChunkGenerator< T >`:



## Public Member Functions

- **ChunkGenerator** (`size_t` size, [GeneratorWrapper< T >](#) generator)
- `std::vector< T > const & get ()` const override
- `bool next ()` override

## Additional Inherited Members

### 5.46.1 Member Function Documentation

#### 5.46.1.1 next()

```

template<typename T >
bool Catch::Generators::ChunkGenerator< T >::next \( \) [inline], [override], [virtual]
  
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.47 Catch::Colour Struct Reference

### Public Types

- enum **Code** {  
**None** = 0 , **White** , **Red** , **Green** ,  
**Blue** , **Cyan** , **Yellow** , **Grey** ,  
**Bright** = 0x10 , **BrightRed** = Bright | Red , **BrightGreen** = Bright | Green , **LightGrey** = Bright | Grey ,  
**BrightWhite** = Bright | White , **BrightYellow** = Bright | Yellow , **FileName** = LightGrey , **Warning** = Bright↵  
Yellow ,  
**ResultError** = BrightRed , **ResultSuccess** = BrightGreen , **ResultExpectedFailure** = Warning , **Error** =  
BrightRed ,  
**Success** = Green , **OriginalExpression** = Cyan , **ReconstructedExpression** = BrightYellow , **Secondary**↵  
**Text** = LightGrey ,  
**Headers** = White }

The documentation for this struct was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.48 Catch::ColourImpl::ColourGuard Class Reference

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- [ColourGuard](#) (Colour::Code code, [ColourImpl](#) const \*colour)  
*Does **not** engage the guard/start the colour.*
- ColourGuard** ([ColourGuard](#) const &rhs)=delete
- [ColourGuard](#) & **operator=** ([ColourGuard](#) const &rhs)=delete
- ColourGuard** ([ColourGuard](#) &&rhs) noexcept
- [ColourGuard](#) & **operator=** ([ColourGuard](#) &&rhs) noexcept
- [~ColourGuard](#) ()  
*Removes colour if the guard was engaged.*
- [ColourGuard](#) & **engage** (std::ostream &stream) &
- [ColourGuard](#) && **engage** (std::ostream &stream) &&

### Friends

- std::ostream & **operator<<** (std::ostream &lhs, [ColourGuard](#) &guard)  
*Engages the guard and starts using colour.*
- std::ostream & **operator<<** (std::ostream &lhs, [ColourGuard](#) &&guard)  
*Engages the guard and starts using colour.*

### 5.48.1 Detailed Description

RAII wrapper around writing specific colour of text using specific colour impl into a stream.

## 5.48.2 Member Function Documentation

### 5.48.2.1 `engage()` [1/2]

```
ColourGuard& Catch::ColourImpl::ColourGuard::engage (
    std::ostream & stream ) &
```

Explicitly engages colour for given stream.

The API based on operator<< should be preferred.

### 5.48.2.2 `engage()` [2/2]

```
ColourGuard&& Catch::ColourImpl::ColourGuard::engage (
    std::ostream & stream ) &&
```

Explicitly engages colour for given stream.

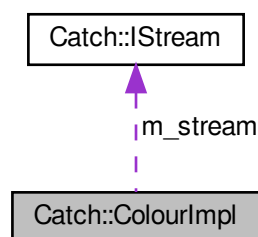
The API based on operator<< should be preferred.

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.49 Catch::ColourImpl Class Reference

Collaboration diagram for Catch::ColourImpl:



## Classes

- class `ColourGuard`

## Public Member Functions

- **ColourImpl** ([IStream](#) \*stream)
- **ColourGuard** [guardColour](#) ([Colour::Code](#) colourCode)

## Protected Attributes

- [IStream](#) \* [m\\_stream](#)  
The associated stream of this [ColourImpl](#) instance.

## 5.49.1 Member Function Documentation

### 5.49.1.1 [guardColour\(\)](#)

```
ColourGuard Catch::ColourImpl::guardColour (
    Colour::Code colourCode )
```

Creates a guard object for given colour and this colour impl

**Important:** the guard starts disengaged, and has to be engaged explicitly.

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.50 Catch::TextFlow::Column Class Reference

```
#include <catch_amalgamated.hpp>
```

## Classes

- class [const\\_iterator](#)

## Public Types

- using **iterator** = [const\\_iterator](#)

## Public Member Functions

- **Column** (std::string const &text)
- [Column](#) & **width** (size\_t newWidth)
- [Column](#) & **indent** (size\_t newIndent)
- [Column](#) & **initialIndent** (size\_t newIndent)
- size\_t **width** () const
- [const\\_iterator](#) **begin** () const
- [const\\_iterator](#) **end** () const
- [Columns](#) **operator+** ([Column](#) const &other)

## Friends

- `std::ostream & operator<< (std::ostream &os, Column const &col)`

### 5.50.1 Detailed Description

Represents a column of text with specific width and indentation

When written out to a stream, it will perform linebreaking of the provided text so that the written lines fit within target width.

The documentation for this class was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.51 `Catch::TextFlow::Columns` Class Reference

### Classes

- class [iterator](#)

### Public Types

- using `const_iterator` = [iterator](#)

### Public Member Functions

- [iterator](#) `begin` () const
- [iterator](#) `end` () const
- [Columns](#) & `operator+=` ([Column](#) const &col)
- [Columns](#) `operator+` ([Column](#) const &col)

## Friends

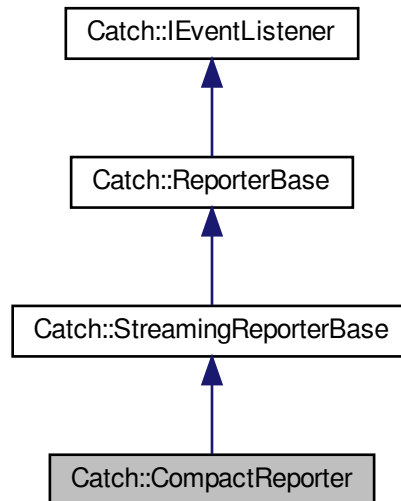
- `std::ostream & operator<< (std::ostream &os, Columns const &cols)`

The documentation for this class was generated from the following file:

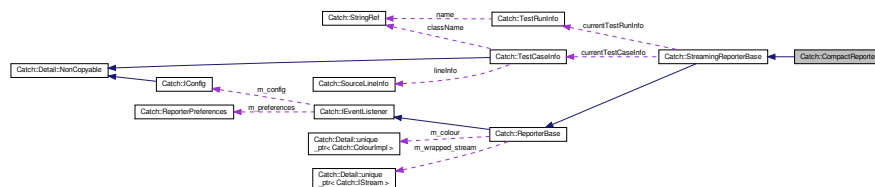
- `include/catch2/catch\_amalgamated.hpp`

## 5.52 Catch::CompactReporter Class Reference

Inheritance diagram for Catch::CompactReporter:



Collaboration diagram for Catch::CompactReporter:



### Public Member Functions

- void `noMatchingTestCases` (`StringRef` unmatchedSpec) override  
*Called when no test cases match provided test spec.*
- void `testRunStarting` (`TestRunInfo` const &\_testInfo) override
- void `assertionEnded` (`AssertionStats` const &\_assertionStats) override  
*Called after assertion was fully evaluated.*
- void `sectionEnded` (`SectionStats` const &\_sectionStats) override  
*Called after a SECTION has finished running.*
- void `testRunEnded` (`TestRunStats` const &\_testRunStats) override

### Static Public Member Functions

- static std::string `getDescription` ()

## Additional Inherited Members

### 5.52.1 Member Function Documentation

#### 5.52.1.1 testRunEnded()

```
void Catch::CompactReporter::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Reimplemented from [Catch::StreamingReporterBase](#).

#### 5.52.1.2 testRunStarting()

```
void Catch::CompactReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Reimplemented from [Catch::StreamingReporterBase](#).

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.53 Catch::Benchmark::Detail::CompleterInvoker< Result > Struct Template Reference

### Static Public Member Functions

- `template<typename Fun , typename... Args>`  
static Result **invoke** (Fun &&fun, Args &&... args)

The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)



## 5.54 Catch::Benchmark::Detail::CompleteInvoker< void > Struct Reference

### Static Public Member Functions

- `template<typename Fun , typename... Args>`  
`static CompleteType_t< void > invoke (Fun &&fun, Args &&... args)`

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.55 Catch::Benchmark::Detail::CompleteType< T > Struct Template Reference

### Public Types

- `using type = T`

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.56 Catch::Benchmark::Detail::CompleteType< void > Struct Reference

### Classes

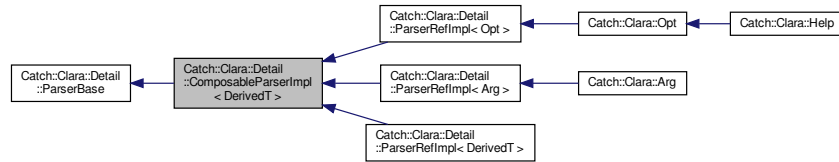
- struct `type`

The documentation for this struct was generated from the following file:

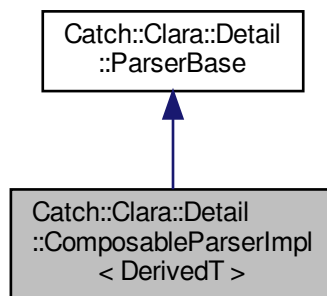
- `include/catch2/catch_amalgamated.hpp`

## 5.57 Catch::Clara::Detail::ComposableParserImpl< DerivedT > Class Template Reference

Inheritance diagram for Catch::Clara::Detail::ComposableParserImpl< DerivedT >:



Collaboration diagram for Catch::Clara::Detail::ComposableParserImpl< DerivedT >:



### Public Member Functions

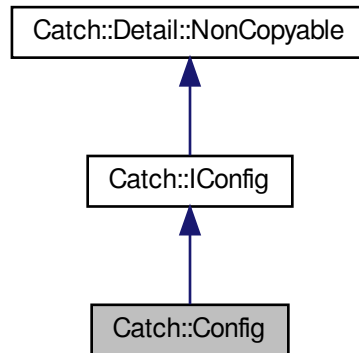
- `template<typename T >`  
`auto operator| (T const &other) const -> Parser`
- `template<typename T >`  
`Parser operator| (T const &other) const`

The documentation for this class was generated from the following file:

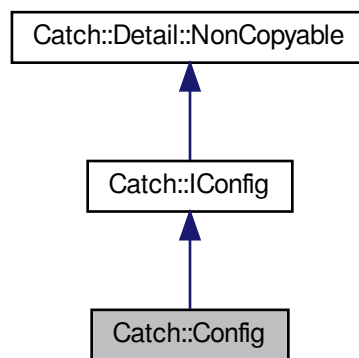
- `include/catch2/catch\_amalgamated.hpp`

## 5.58 Catch::Config Class Reference

Inheritance diagram for Catch::Config:



Collaboration diagram for Catch::Config:



### Public Member Functions

- **Config** ([ConfigData](#) const &data)
- bool **listTests** () const
- bool **listTags** () const
- bool **listReporters** () const
- bool **listListeners** () const
- std::vector< [ReporterSpec](#) > const & **getReporterSpecs** () const
- std::vector< [ProcessedReporterSpec](#) > const & **getProcessedReporterSpecs** () const

- `std::vector< std::string > const & getTestsOrTags ()` const override
- `std::vector< std::string > const & getSectionsToRun ()` const override
- `TestSpec const & testSpec ()` const override
- `bool hasTestFilters ()` const override
- `bool showHelp ()` const
- `bool allowThrows ()` const override
- `StringRef name ()` const override
- `bool includeSuccessfulResults ()` const override
- `bool warnAboutMissingAssertions ()` const override
- `bool warnAboutUnmatchedTestSpecs ()` const override
- `bool zeroTestsCountAsSuccess ()` const override
- `ShowDurations showDurations ()` const override
- `double minDuration ()` const override
- `TestRunOrder runOrder ()` const override
- `uint32_t rngSeed ()` const override
- `unsigned int shardCount ()` const override
- `unsigned int shardIndex ()` const override
- `ColourMode defaultColourMode ()` const override
- `bool shouldDebugBreak ()` const override
- `int abortAfter ()` const override
- `bool showInvisibles ()` const override
- `Verbosity verbosity ()` const override
- `bool skipBenchmarks ()` const override
- `bool benchmarkNoAnalysis ()` const override
- `unsigned int benchmarkSamples ()` const override
- `double benchmarkConfidenceInterval ()` const override
- `unsigned int benchmarkResamples ()` const override
- `std::chrono::milliseconds benchmarkWarmupTime ()` const override

The documentation for this class was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.59 Catch::ConfigData Struct Reference

### Public Attributes

- `bool listTests = false`
- `bool listTags = false`
- `bool listReporters = false`
- `bool listListeners = false`
- `bool showSuccessfulTests = false`
- `bool shouldDebugBreak = false`
- `bool noThrow = false`
- `bool showHelp = false`
- `bool showInvisibles = false`
- `bool filenamesAsTags = false`
- `bool libIdentify = false`
- `bool allowZeroTests = false`
- `int abortAfter = -1`
- `uint32_t rngSeed = generateRandomSeed(GenerateFrom::Default)`

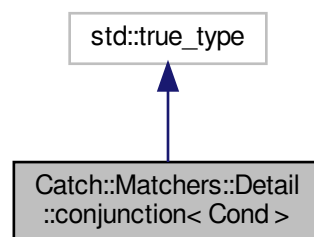
- unsigned int **shardCount** = 1
- unsigned int **shardIndex** = 0
- bool **skipBenchmarks** = false
- bool **benchmarkNoAnalysis** = false
- unsigned int **benchmarkSamples** = 100
- double **benchmarkConfidenceInterval** = 0.95
- unsigned int **benchmarkResamples** = 100000
- std::chrono::milliseconds::rep **benchmarkWarmupTime** = 100
- Verbosity **verbosity** = Verbosity::Normal
- WarnAbout::What **warnings** = WarnAbout::Nothing
- ShowDurations **showDurations** = ShowDurations::DefaultForReporter
- double **minDuration** = -1
- TestRunOrder **runOrder** = TestRunOrder::Declared
- ColourMode **defaultColourMode** = ColourMode::PlatformDefault
- WaitForKeypress::When **waitForKeypress** = WaitForKeypress::Never
- std::string **defaultOutputFilename**
- std::string **name**
- std::string **processName**
- std::vector< ReporterSpec > **reporterSpecifications**
- std::vector< std::string > **testsOrTags**
- std::vector< std::string > **sectionsToRun**

The documentation for this struct was generated from the following file:

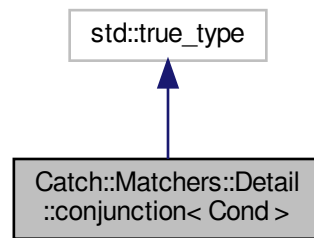
- include/catch2/catch\_amalgamated.hpp

## 5.60 Catch::Matchers::Detail::conjunction< Cond > Struct Template Reference

Inheritance diagram for Catch::Matchers::Detail::conjunction< Cond >:



Collaboration diagram for `Catch::Matchers::Detail::conjunction< Cond >`:

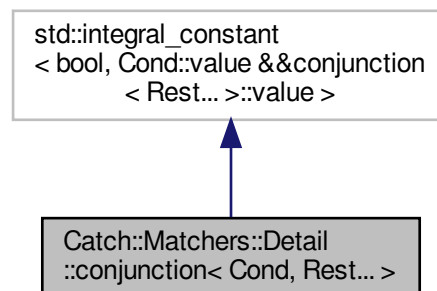


The documentation for this struct was generated from the following file:

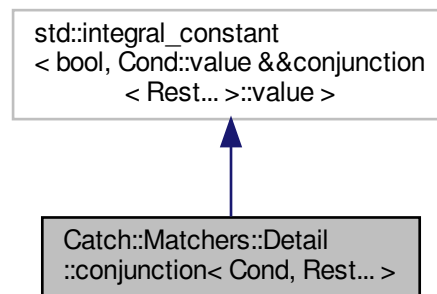
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.61 `Catch::Matchers::Detail::conjunction< Cond, Rest... >` Struct Template Reference

Inheritance diagram for `Catch::Matchers::Detail::conjunction< Cond, Rest... >`:



Collaboration diagram for Catch::Matchers::Detail::conjunction< Cond, Rest... >:

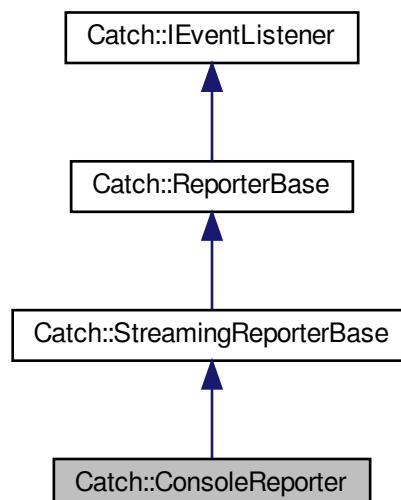


The documentation for this struct was generated from the following file:

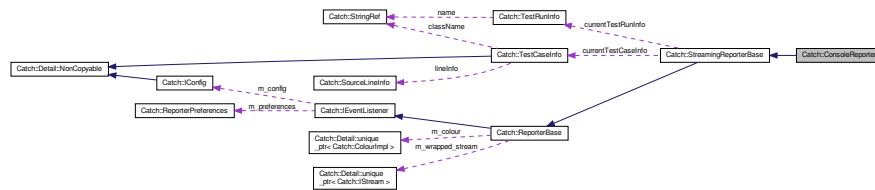
- `include/catch2/catch_amalgamated.hpp`

## 5.62 Catch::ConsoleReporter Class Reference

Inheritance diagram for Catch::ConsoleReporter:



Collaboration diagram for `Catch::ConsoleReporter`:



## Public Member Functions

- **ConsoleReporter** ([ReporterConfig](#) &&config)
- void **noMatchingTestCases** ([StringRef](#) unmatchedSpec) override  
*Called when no test cases match provided test spec.*
- void **reportInvalidTestSpec** ([StringRef](#) arg) override  
*Called for all invalid test specs from the cli.*
- void **assertionStarting** ([AssertionInfo](#) const &) override  
*Called before assertion success/failure is evaluated.*
- void **assertionEnded** ([AssertionStats](#) const &\_assertionStats) override  
*Called after assertion was fully evaluated.*
- void **sectionStarting** ([SectionInfo](#) const &\_sectionInfo) override  
*Called when a SECTION is being entered. Not called for skipped sections.*
- void **sectionEnded** ([SectionStats](#) const &\_sectionStats) override  
*Called after a SECTION has finished running.*
- void **benchmarkPreparing** ([StringRef](#) name) override  
*Called when user-code is being probed before the actual benchmark runs.*
- void **benchmarkStarting** ([BenchmarkInfo](#) const &info) override  
*Called after probe but before the user-code is being benchmarked.*
- void **benchmarkEnded** ([BenchmarkStats](#)<> const &stats) override  
*Called with the benchmark results if benchmark successfully finishes.*
- void **benchmarkFailed** ([StringRef](#) error) override  
*Called if running the benchmarks fails for any reason.*
- void **testCaseEnded** ([TestCaseStats](#) const &\_testCaseStats) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void **testRunEnded** ([TestRunStats](#) const &\_testRunStats) override
- void **testRunStarting** ([TestRunInfo](#) const &\_testRunInfo) override

## Static Public Member Functions

- static std::string **getDescription** ()

## Additional Inherited Members

### 5.62.1 Member Function Documentation



### 5.62.1.1 testRunEnded()

```
void Catch::ConsoleReporter::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Reimplemented from [Catch::StreamingReporterBase](#).

### 5.62.1.2 testRunStarting()

```
void Catch::ConsoleReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Reimplemented from [Catch::StreamingReporterBase](#).

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.63 Catch::TextFlow::Column::const\_iterator Class Reference

```
#include <catch_amalgamated.hpp>
```

### Public Types

- using **difference\_type** = std::ptrdiff\_t
- using **value\_type** = std::string
- using **pointer** = value\_type \*
- using **reference** = value\_type &
- using **iterator\_category** = std::forward\_iterator\_tag

### Public Member Functions

- **const\_iterator** ([Column](#) const &column)
- std::string **operator\*** () const
- [const\\_iterator](#) & **operator++** ()
- [const\\_iterator](#) **operator++** (int)
- bool **operator==** ([const\\_iterator](#) const &other) const
- bool **operator!=** ([const\\_iterator](#) const &other) const

### 5.63.1 Detailed Description

Iterates "lines" in [Column](#) and return sthem

The documentation for this class was generated from the following file:

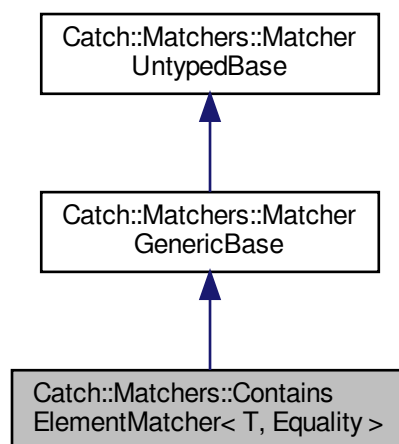
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.64 `Catch::Matchers::ContainsElementMatcher< T, Equality >` Class Template Reference

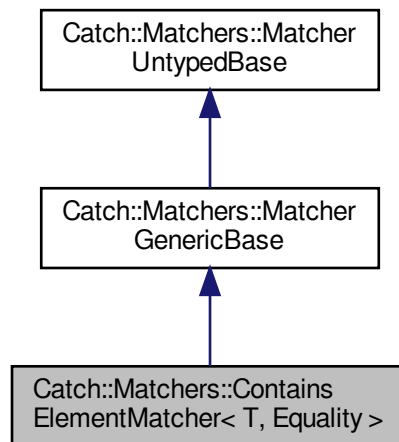
Matcher for checking that an element in range is equal to specific element.

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for `Catch::Matchers::ContainsElementMatcher< T, Equality >`:



Collaboration diagram for Catch::Matchers::ContainsElementMatcher< T, Equality >:



## Public Member Functions

- `template<typename T2 , typename Equality2 >`  
**ContainsElementMatcher** (T2 &&target, Equality2 &&predicate)
- `std::string describe ()` const override
- `template<typename RangeLike >`  
**bool match** (RangeLike &&rng) const

## Additional Inherited Members

### 5.64.1 Detailed Description

```
template<typename T, typename Equality>
class Catch::Matchers::ContainsElementMatcher< T, Equality >
```

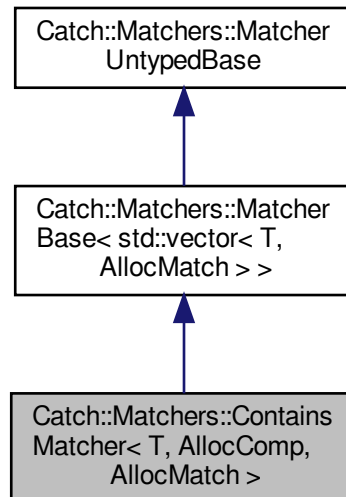
Matcher for checking that an element in range is equal to specific element.

The documentation for this class was generated from the following file:

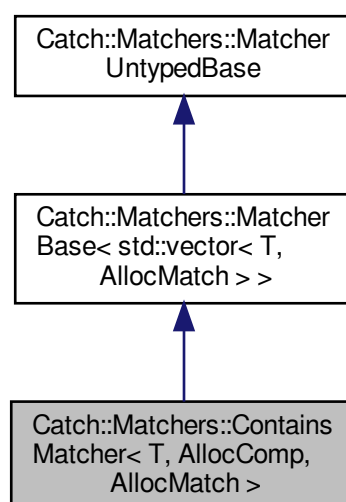
- `include/catch2/catch_amalgamated.hpp`

## 5.65 Catch::Matchers::ContainsMatcher< T, AllocComp, AllocMatch > Class Template Reference

Inheritance diagram for Catch::Matchers::ContainsMatcher< T, AllocComp, AllocMatch >:



Collaboration diagram for Catch::Matchers::ContainsMatcher< T, AllocComp, AllocMatch >:



## Public Member Functions

- **ContainsMatcher** (std::vector< T, AllocComp > const &comparator)
- bool **match** (std::vector< T, AllocMatch > const &v) const override
- std::string **describe** () const override

## Additional Inherited Members

The documentation for this class was generated from the following file:

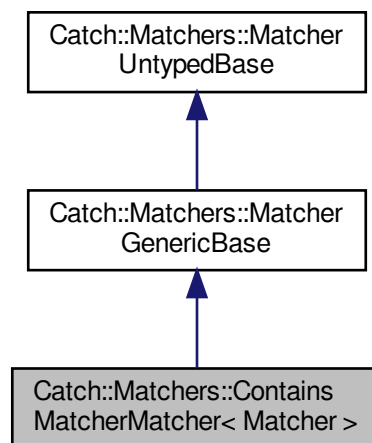
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.66 Catch::Matchers::ContainsMatcherMatcher< Matcher > Class Template Reference

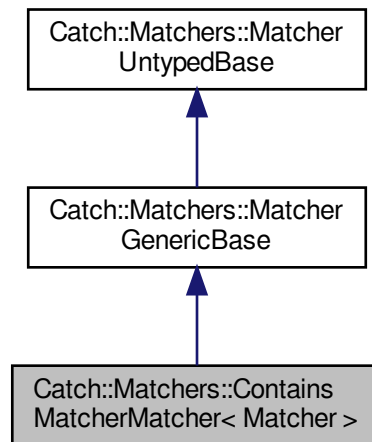
Meta-matcher for checking that an element in a range matches a specific matcher.

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for Catch::Matchers::ContainsMatcherMatcher< Matcher >:



Collaboration diagram for `Catch::Matchers::ContainsMatcherMatcher< Matcher >`:



## Public Member Functions

- **ContainsMatcherMatcher** (Matcher matcher)
- `template<typename RangeLike >`  
`bool match` (RangeLike &&rng) const
- `std::string describe` () const override

## Additional Inherited Members

### 5.66.1 Detailed Description

```
template<typename Matcher>
class Catch::Matchers::ContainsMatcherMatcher< Matcher >
```

Meta-matcher for checking that an element in a range matches a specific matcher.

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.67 Catch::Counts Struct Reference

### Public Member Functions

- `Counts operator-` (Counts const &other) const
- `Counts & operator+=` (Counts const &other)
- `std::uint64_t total` () const
- `bool allPassed` () const
- `bool allOk` () const

## Public Attributes

- `std::uint64_t` **passed** = 0
- `std::uint64_t` **failed** = 0
- `std::uint64_t` **failedButOk** = 0

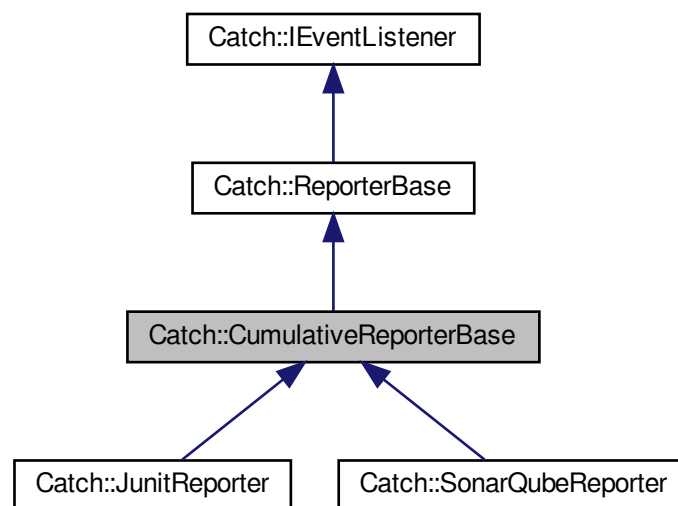
The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

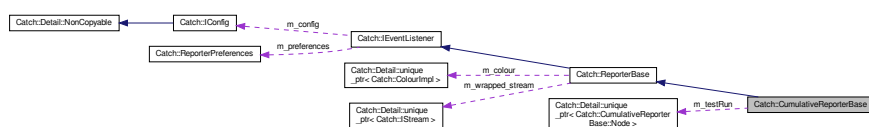
## 5.68 Catch::CumulativeReporterBase Class Reference

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for Catch::CumulativeReporterBase:



Collaboration diagram for Catch::CumulativeReporterBase:



## Classes

- struct [Node](#)
- struct [SectionNode](#)

## Public Types

- using **TestCaseNode** = [Node](#)< [TestCaseStats](#), [SectionNode](#) >
- using **TestRunNode** = [Node](#)< [TestRunStats](#), [TestCaseNode](#) >

## Public Member Functions

- void [benchmarkPreparing](#) ([StringRef](#)) override  
*Called when user-code is being probed before the actual benchmark runs.*
- void [benchmarkStarting](#) ([BenchmarkInfo](#) const &) override  
*Called after probe but before the user-code is being benchmarked.*
- void [benchmarkEnded](#) ([BenchmarkStats](#)<> const &benchmarkStats) override  
*Called with the benchmark results if benchmark successfully finishes.*
- void [benchmarkFailed](#) ([StringRef](#)) override  
*Called if running the benchmarks fails for any reason.*
- void [noMatchingTestCases](#) ([StringRef](#)) override  
*Called when no test cases match provided test spec.*
- void [reportInvalidTestSpec](#) ([StringRef](#)) override  
*Called for all invalid test specs from the cli.*
- void [fatalErrorEncountered](#) ([StringRef](#)) override  
*Called if a fatal error (signal/structured exception) occurred.*
- void [testRunStarting](#) ([TestRunInfo](#) const &) override
- void [testCaseStarting](#) ([TestCaseInfo](#) const &) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testCasePartialStarting](#) ([TestCaseInfo](#) const &, uint64\_t) override  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- void [sectionStarting](#) ([SectionInfo](#) const &sectionInfo) override  
*Called when a SECTION is being entered. Not called for skipped sections.*
- void [assertionStarting](#) ([AssertionInfo](#) const &) override  
*Called before assertion success/failure is evaluated.*
- void [assertionEnded](#) ([AssertionStats](#) const &assertionStats) override  
*Called after assertion was fully evaluated.*
- void [sectionEnded](#) ([SectionStats](#) const &sectionStats) override  
*Called after a SECTION has finished running.*
- void [testCasePartialEnded](#) ([TestCaseStats](#) const &, uint64\_t) override  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- void [testCaseEnded](#) ([TestCaseStats](#) const &testCaseStats) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testRunEnded](#) ([TestRunStats](#) const &testRunStats) override
- virtual void [testRunEndedCumulative](#) ()=0  
*Customization point: called after last test finishes (testRunEnded has been handled)*
- void [skipTest](#) ([TestCaseInfo](#) const &) override  
*Called with test cases that are skipped due to the test run aborting.*
- **ReporterBase** ([ReporterConfig](#) &&config)

## Protected Attributes

- bool [m\\_shouldStoreSuccessfulAssertions](#) = true  
*Should the cumulative base store the assertion expansion for succesful assertions?*
- bool [m\\_shouldStoreFailedAssertions](#) = true  
*Should the cumulative base store the assertion expansion for failed assertions?*
- [Detail::unique\\_ptr](#)< [TestRunNode](#) > [m\\_testRun](#)  
*The root node of the test run tree.*



### 5.68.1 Detailed Description

Utility base for reporters that need to handle all results at once

It stores tree of all test cases, sections and assertions, and after the test run is finished, calls into `testRunEndedCumulative` to pass the control to the deriving class.

If you are deriving from this class and override any testing related member functions, you should first call into the base's implementation to avoid breaking the tree construction.

Due to the way this base functions, it has to expand assertions up-front, even if they are later unused (e.g. because the deriving reporter does not report successful assertions, or because the deriving reporter does not use assertion expansion at all). Derived classes can use two customization points, `m_shouldStoreSuccessfulAssertions` and `m_shouldStoreFailedAssertions`, to disable the expansion and gain extra performance. **Accessing the assertion expansions if it wasn't stored is UB.**

### 5.68.2 Member Function Documentation

#### 5.68.2.1 testRunEnded()

```
void Catch::CumulativeReporterBase::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Implements [Catch::IEventListener](#).

#### 5.68.2.2 testRunStarting()

```
void Catch::CumulativeReporterBase::testRunStarting (
    TestRunInfo const & testRunInfo ) [inline], [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Implements [Catch::IEventListener](#).

Reimplemented in [Catch::SonarQubeReporter](#), and [Catch::JUnitReporter](#).

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.69 DataBase< TSeq > Class Template Reference

Statistical data about the process.

```
#include <database-bones.hpp>
```

### Public Member Functions

- **DataBase** ([Model](#)< TSeq > &m)
- void [record\\_variant](#) ([Virus](#)< TSeq > &v)  
*Registering a new variant.*
- void [record\\_tool](#) ([Tool](#)< TSeq > &t)
- void [set\\_seq\\_hasher](#) (std::function< std::vector< int >(TSeq)> fun)
- void [set\\_model](#) ([Model](#)< TSeq > &m)
- [Model](#)< TSeq > \* [get\\_model](#) ()
- void [record](#) ()
- const std::vector< TSeq > & [get\\_sequence](#) () const
- const std::vector< int > & [get\\_nexposed](#) () const
- size\_t [size](#) () const
- void [write\\_data](#) (std::string fn\_variant\_info, std::string fn\_variant\_hist, std::string fn\_tool\_info, std::string fn\_tool\_hist, std::string fn\_total\_hist, std::string fn\_transmission, std::string fn\_transition) const
- void [record\\_transmission](#) (int i, int j, int variant, int i\_expo\_date)
- size\_t [get\\_n\\_variants](#) () const
- size\_t [get\\_n\\_tools](#) () const
- void [reset](#) ()
- void [set\\_user\\_data](#) (std::vector< std::string > names)
- void [add\\_user\\_data](#) (std::vector< epiworld\_double > x)
- void [add\\_user\\_data](#) (unsigned int j, epiworld\_double x)
- [UserData](#)< TSeq > & [get\\_user\\_data](#) ()

### Get recorded information from the model

#### Parameters

what	<i>std::string, The status, e.g., 0, 1, 2, ...</i>
------	--

#### Returns

*In [get\\_today\\_total](#), the current counts of what.*

*In [get\\_today\\_variant](#), the current counts of what for each variant.*

*In [get\\_hist\\_total](#), the time series of what*

*In [get\\_hist\\_variant](#), the time series of what for each variant.*

*In [get\\_hist\\_total\\_date](#) and [get\\_hist\\_variant\\_date](#) the corresponding dates*

- int [get\\_today\\_total](#) (std::string what) const
- int [get\\_today\\_total](#) (epiworld\_fast\_uint what) const
- void [get\\_today\\_total](#) (std::vector< std::string > \*status=nullptr, std::vector< int > \*counts=nullptr) const
- void [get\\_today\\_variant](#) (std::vector< std::string > &status, std::vector< int > &id, std::vector< int > &counts) const
- void [get\\_hist\\_total](#) (std::vector< int > \*date, std::vector< std::string > \*status, std::vector< int > \*counts) const
- void [get\\_hist\\_variant](#) (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &status, std::vector< int > &counts) const

- MapVec\_type< int, int > [reproductive\\_number](#) () const  
*Computes the reproductive number of each case.*
- void **reproductive\_number** (std::string fn) const

## Friends

- class **Model**< TSeq >
- void **default\_add\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_add\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)

## 5.69.1 Detailed Description

```
template<typename TSeq>
class DataBase< TSeq >
```

Statistical data about the process.

### Template Parameters

<i>TSeq</i>	
-------------	--

## 5.69.2 Member Function Documentation

### 5.69.2.1 record\_variant()

```
template<typename TSeq >
void DataBase< TSeq >::record_variant (
    Virus< TSeq > & v ) [inline]
```

Registering a new variant.

### Parameters

<i>v</i>	Pointer to the new variant. Since variants are originated in the agent, the numbers simply move around. From the parent variant to the new variant. And the total number of infected does not change.
----------	---

### 5.69.2.2 reproductive\_number()

```
template<typename TSeq >
MapVec_type< int, int > DataBase< TSeq >::reproductive_number [inline]
```

Computes the reproductive number of each case.

#### Parameters

<i>fn</i>	File where to write out the reproductive number.
-----------	--

The documentation for this class was generated from the following files:

- include/epiworld/database-bones.hpp
- include/epiworld/database-meat.hpp

## 5.70 Catch::Decomposer Struct Reference

### Friends

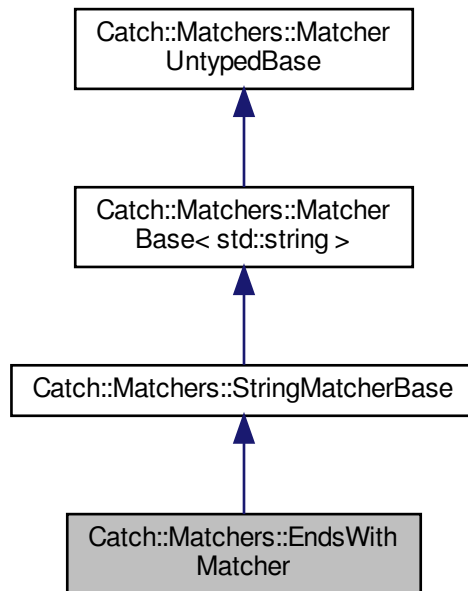
- template<typename T, std::enable\_if\_t<!std::is\_arithmetic< std::remove\_reference\_t< T >>::value, int > = 0>  
auto **operator**<= (Decomposer &&, T &&lhs) -> ExprLhs< T const & >
- template<typename T, std::enable\_if\_t< std::is\_arithmetic< T >::value, int > = 0>  
auto **operator**<= (Decomposer &&, T value) -> ExprLhs< T >

The documentation for this struct was generated from the following file:

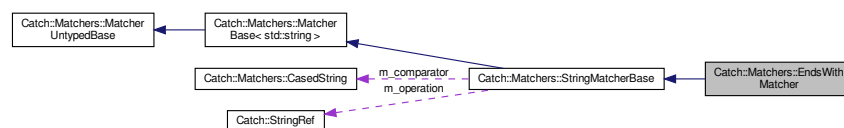
- include/catch2/catch\_amalgamated.hpp

## 5.71 Catch::Matchers::EndsWithMatcher Class Reference

Inheritance diagram for Catch::Matchers::EndsWithMatcher:



Collaboration diagram for Catch::Matchers::EndsWithMatcher:



### Public Member Functions

- **EndsWithMatcher** ([CasedString](#) const &comparator)
- **bool match** (std::string const &source) const override

### Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.72 Entity< TSeq > Class Template Reference

### Public Member Functions

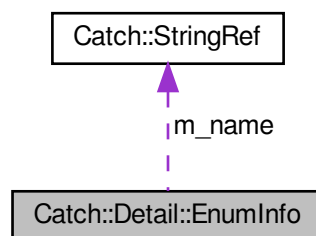
- void **add\_agent** ([Agent](#)< TSeq > &p)
- void **add\_agent** ([Agent](#)< TSeq > \*p)
- void **rm\_agent** (size\_t idx)
- size\_t **size** () const noexcept
- void **set\_location** (std::vector< epiworld\_double > loc)
- std::vector< epiworld\_double > & **get\_location** ()
- std::vector< [Agent](#)< TSeq > \* >::iterator **begin** ()
- std::vector< [Agent](#)< TSeq > \* >::iterator **end** ()
- std::vector< [Agent](#)< TSeq > \* >::const\_iterator **begin** () const
- std::vector< [Agent](#)< TSeq > \* >::const\_iterator **end** () const

The documentation for this class was generated from the following file:

- include/epiworld/entity-bones.hpp

## 5.73 Catch::Detail::EnumInfo Struct Reference

Collaboration diagram for Catch::Detail::EnumInfo:



### Public Member Functions

- [StringRef](#) **lookup** (int value) const

### Public Attributes

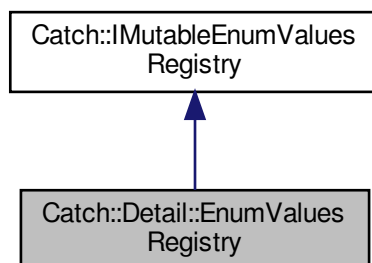
- [StringRef](#) **m\_name**
- std::vector< std::pair< int, [StringRef](#) > > **m\_values**

The documentation for this struct was generated from the following file:

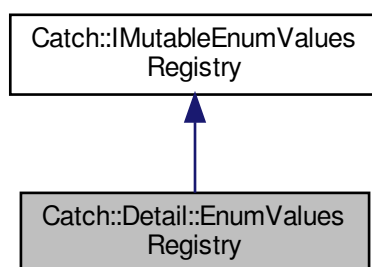
- include/catch2/catch\_amalgamated.hpp

## 5.74 Catch::Detail::EnumValuesRegistry Class Reference

Inheritance diagram for Catch::Detail::EnumValuesRegistry:



Collaboration diagram for Catch::Detail::EnumValuesRegistry:



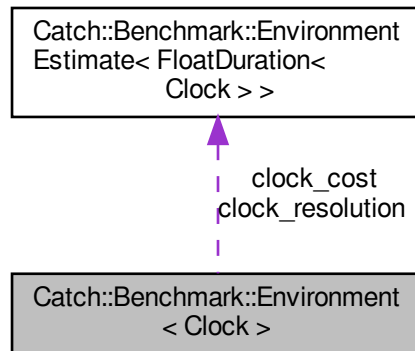
### Additional Inherited Members

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.75 Catch::Benchmark::Environment< Clock > Struct Template Reference

Collaboration diagram for Catch::Benchmark::Environment< Clock >:



### Public Types

- using **clock\_type** = Clock

### Public Attributes

- [EnvironmentEstimate](#)< FloatDuration< Clock > > **clock\_resolution**
- [EnvironmentEstimate](#)< FloatDuration< Clock > > **clock\_cost**

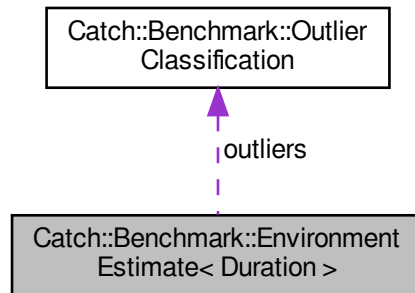
The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



## 5.76 Catch::Benchmark::EnvironmentEstimate< Duration > Struct Template Reference

Collaboration diagram for Catch::Benchmark::EnvironmentEstimate< Duration >:



### Public Member Functions

- `template<typename Duration2 >`  
`operator EnvironmentEstimate< Duration2 > () const`

### Public Attributes

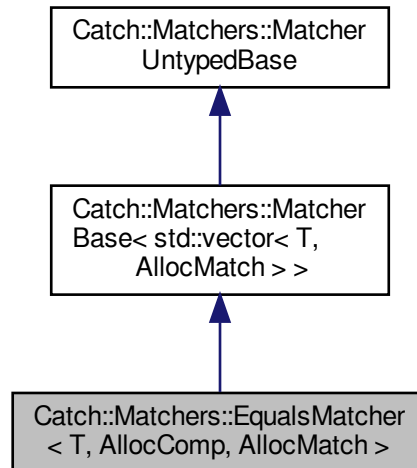
- Duration **mean**
- [OutlierClassification](#) **outliers**

The documentation for this struct was generated from the following file:

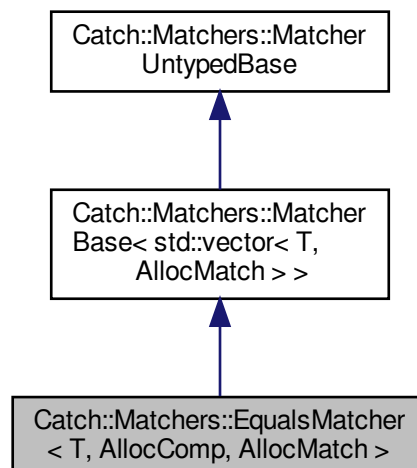
- `include/catch2/catch\_amalgamated.hpp`

## 5.77 Catch::Matchers::EqualsMatcher< T, AllocComp, AllocMatch > Class Template Reference

Inheritance diagram for Catch::Matchers::EqualsMatcher< T, AllocComp, AllocMatch >:



Collaboration diagram for Catch::Matchers::EqualsMatcher< T, AllocComp, AllocMatch >:



### Public Member Functions

- **EqualsMatcher** (std::vector< T, AllocComp > const &comparator)

- bool **match** (std::vector< T, AllocMatch > const &v) const override
- std::string **describe** () const override

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.78 Catch::ErrnoGuard Class Reference

```
#include <catch_amalgamated.hpp>
```

### 5.78.1 Detailed Description

Simple RAII class that stores the value of `errno` at construction and restores it at destruction.

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.79 Catch::Benchmark::Estimate< Duration > Struct Template Reference

### Public Member Functions

- template<typename Duration2 >  
**operator Estimate**< **Duration2** > () const

### Public Attributes

- Duration **point**
- Duration **lower\_bound**
- Duration **upper\_bound**
- double **confidence\_interval**

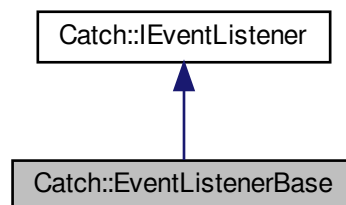
The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

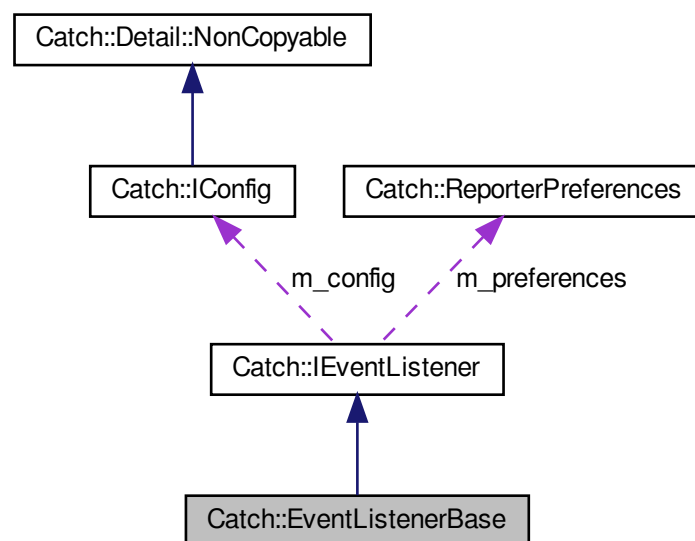
## 5.80 Catch::EventListenerBase Class Reference

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for Catch::EventListenerBase:



Collaboration diagram for Catch::EventListenerBase:



### Public Member Functions

- void [reportInvalidTestSpec](#) ([StringRef](#) unmatchedSpec) override  
*Called for all invalid test specs from the cli.*
- void [fatalErrorEncountered](#) ([StringRef](#) error) override  
*Called if a fatal error (signal/structured exception) occurred.*

- void [benchmarkPreparing](#) ([StringRef](#) name) override  
*Called when user-code is being probed before the actual benchmark runs.*
- void [benchmarkStarting](#) ([BenchmarkInfo](#) const &benchmarkInfo) override  
*Called after probe but before the user-code is being benchmarked.*
- void [benchmarkEnded](#) ([BenchmarkStats](#)<> const &benchmarkStats) override  
*Called with the benchmark results if benchmark successfully finishes.*
- void [benchmarkFailed](#) ([StringRef](#) error) override  
*Called if running the benchmarks fails for any reason.*
- void [assertionStarting](#) ([AssertionInfo](#) const &assertionInfo) override  
*Called before assertion success/failure is evaluated.*
- void [assertionEnded](#) ([AssertionStats](#) const &assertionStats) override  
*Called after assertion was fully evaluated.*
- void [listReporters](#) (std::vector< [ReporterDescription](#) > const &descriptions) override  
*Writes out information about provided reporters using reporter-specific format.*
- void [listListeners](#) (std::vector< [ListenerDescription](#) > const &descriptions) override  
*Writes out the provided listeners descriptions using reporter-specific format.*
- void [listTests](#) (std::vector< [TestCaseHandle](#) > const &tests) override  
*Writes out information about provided tests using reporter-specific format.*
- void [listTags](#) (std::vector< [TagInfo](#) > const &tagInfos) override  
*Writes out information about the provided tags using reporter-specific format.*
- void [noMatchingTestCases](#) ([StringRef](#) unmatchedSpec) override  
*Called when no test cases match provided test spec.*
- void [testRunStarting](#) ([TestRunInfo](#) const &testRunInfo) override
- void [testCaseStarting](#) ([TestCaseInfo](#) const &testInfo) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testCasePartialStarting](#) ([TestCaseInfo](#) const &testInfo, uint64\_t partNumber) override  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- void [sectionStarting](#) ([SectionInfo](#) const &sectionInfo) override  
*Called when a SECTION is being entered. Not called for skipped sections.*
- void [sectionEnded](#) ([SectionStats](#) const &sectionStats) override  
*Called after a SECTION has finished running.*
- void [testCasePartialEnded](#) ([TestCaseStats](#) const &testCaseStats, uint64\_t partNumber) override  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- void [testCaseEnded](#) ([TestCaseStats](#) const &testCaseStats) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testRunEnded](#) ([TestRunStats](#) const &testRunStats) override
- void [skipTest](#) ([TestCaseInfo](#) const &testInfo) override  
*Called with test cases that are skipped due to the test run aborting.*
- [IEventListener](#) ([IConfig](#) const \*config)

## Additional Inherited Members

### 5.80.1 Detailed Description

Base class to simplify implementing listeners.

Provides empty default implementation for all [IEventListener](#) member functions, so that a listener implementation can pick which member functions it actually cares about.

## 5.80.2 Member Function Documentation

### 5.80.2.1 testRunEnded()

```
void Catch::EventListenerBase::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Implements [Catch::IEventListener](#).

### 5.80.2.2 testRunStarting()

```
void Catch::EventListenerBase::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Implements [Catch::IEventListener](#).

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.81 Catch::EventListenerFactory Class Reference

### Public Member Functions

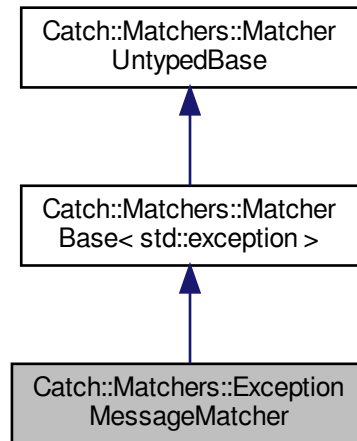
- virtual [IEventListenerPtr](#) **create** ([IConfig](#) const \*config) const =0
- virtual [StringRef](#) **getName** () const =0  
*Return a meaningful name for the listener, e.g. its type name.*
- virtual std::string **getDescription** () const =0  
*Return listener's description if available.*

The documentation for this class was generated from the following file:

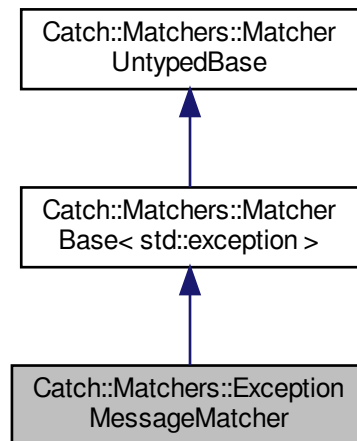
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.82 Catch::Matchers::ExceptionMessageMatcher Class Reference

Inheritance diagram for Catch::Matchers::ExceptionMessageMatcher:



Collaboration diagram for Catch::Matchers::ExceptionMessageMatcher:



### Public Member Functions

- **ExceptionMessageMatcher** (std::string const &message)
- bool **match** (std::exception const &ex) const override
- std::string **describe** () const override

## Additional Inherited Members

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.83 Catch::ExceptionTranslatorRegistrar Class Reference

### Public Member Functions

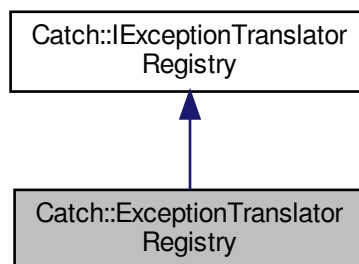
- `template<typename T >`  
**ExceptionTranslatorRegistrar** (std::string(\*translateFunction)(T const &))

The documentation for this class was generated from the following file:

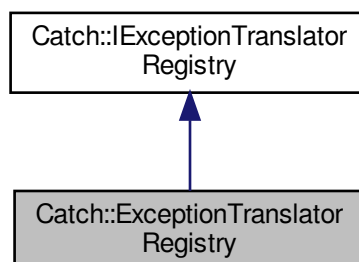
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.84 Catch::ExceptionTranslatorRegistry Class Reference

Inheritance diagram for Catch::ExceptionTranslatorRegistry:



Collaboration diagram for Catch::ExceptionTranslatorRegistry:





## Public Member Functions

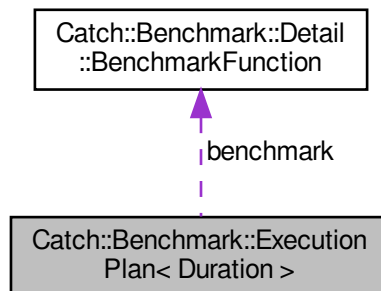
- void **registerTranslator** ([Detail::unique\\_ptr< IExceptionTranslator >](#) &&translator)
- std::string **translateActiveException** () const override
- std::string **tryTranslators** () const

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.85 Catch::Benchmark::ExecutionPlan< Duration > Struct Template Reference

Collaboration diagram for Catch::Benchmark::ExecutionPlan< Duration >:



## Public Member Functions

- template<typename Duration2 >  
**operator ExecutionPlan< Duration2 >** () const
- template<typename Clock >  
std::vector< FloatDuration< Clock > > **run** (const [IConfig](#) &cfg, [Environment](#)< FloatDuration< Clock > > env) const

## Public Attributes

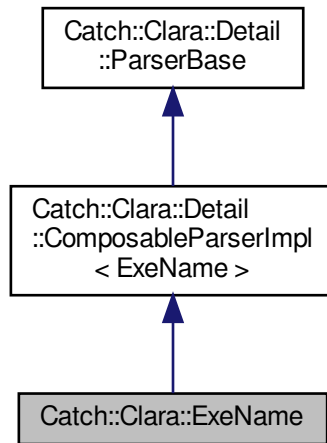
- int **iterations\_per\_sample**
- Duration **estimated\_duration**
- [Detail::BenchmarkFunction](#) **benchmark**
- Duration **warmup\_time**
- int **warmup\_iterations**

The documentation for this struct was generated from the following file:

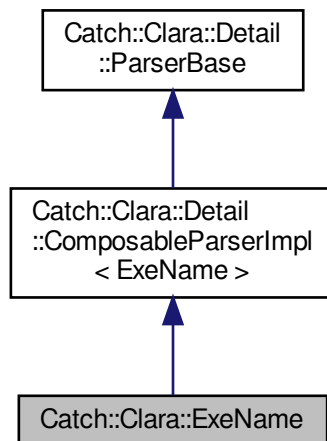
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.86 Catch::Clara::ExeName Class Reference

Inheritance diagram for Catch::Clara::ExeName:



Collaboration diagram for Catch::Clara::ExeName:



### Public Member Functions

- **ExeName** (std::string &ref)
- template<typename LambdaT >  
**ExeName** (LambdaT const &lambda)

- [Detail::InternalParseResult](#) **parse** (std::string const &, [Detail::TokenStream](#) const &tokens) const override
- std::string const & **name** () const
- [Detail::ParserResult](#) **set** (std::string const &newName)

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.87 Catch::ExprLhs< LhsT > Class Template Reference

### Public Member Functions

- **ExprLhs** (LhsT lhs)
- auto **makeUnaryExpr** () const -> [UnaryExpr](#)< LhsT >

### Friends

- template<typename RhsT, std::enable\_if\_t<!std::is\_arithmetic< std::remove\_reference\_t< RhsT >>::value, int > = 0>  
auto **operator==** ([ExprLhs](#) &&lhs, RhsT &&rhs) -> [BinaryExpr](#)< LhsT, RhsT const & >
- template<typename RhsT, std::enable\_if\_t< std::is\_arithmetic< RhsT >::value, int > = 0>  
auto **operator==** ([ExprLhs](#) &&lhs, RhsT rhs) -> [BinaryExpr](#)< LhsT, RhsT >
- template<typename RhsT, std::enable\_if\_t<!std::is\_arithmetic< std::remove\_reference\_t< RhsT >>::value, int > = 0>  
auto **operator!=** ([ExprLhs](#) &&lhs, RhsT &&rhs) -> [BinaryExpr](#)< LhsT, RhsT const & >
- template<typename RhsT, std::enable\_if\_t< std::is\_arithmetic< RhsT >::value, int > = 0>  
auto **operator!=** ([ExprLhs](#) &&lhs, RhsT rhs) -> [BinaryExpr](#)< LhsT, RhsT >
- template<typename RhsT >  
auto **operator&&** ([ExprLhs](#) &&, RhsT &&) -> [BinaryExpr](#)< LhsT, RhsT const & >
- template<typename RhsT >  
auto **operator||** ([ExprLhs](#) &&, RhsT &&) -> [BinaryExpr](#)< LhsT, RhsT const & >

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.88 Catch::Clara::Detail::fake\_arg Struct Reference

### Public Member Functions

- template<typename T >  
**operator T** ()

The documentation for this struct was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.89 Catch::FatalConditionHandler Class Reference

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- void **engage** ()
- void **disengage** () noexcept

#### 5.89.1 Detailed Description

Wrapper for platform-specific fatal error (signals/SEH) handlers

Tries to be cooperative with other handlers, and not step over other handlers. This means that unknown structured exceptions are passed on, previous signal handlers are called, and so on.

Can only be instantiated once, and assumes that once a signal is caught, the binary will end up terminating. Thus, there

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.90 Catch::FatalConditionHandlerGuard Class Reference

Simple RAII guard for (dis)engaging the [FatalConditionHandler](#).

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- **FatalConditionHandlerGuard** ([FatalConditionHandler](#) \*handler)

#### 5.90.1 Detailed Description

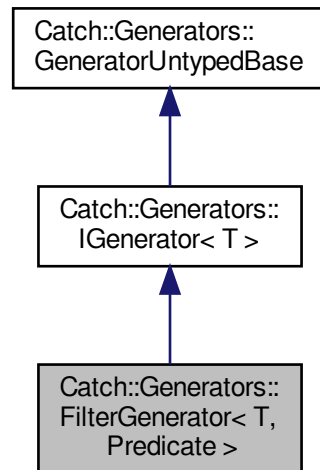
Simple RAII guard for (dis)engaging the [FatalConditionHandler](#).

The documentation for this class was generated from the following file:

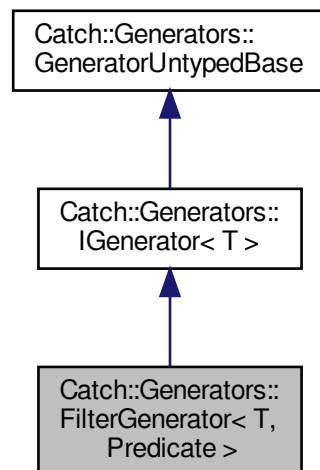
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.91 Catch::Generators::FilterGenerator< T, Predicate > Class Template Reference

Inheritance diagram for Catch::Generators::FilterGenerator< T, Predicate >:



Collaboration diagram for Catch::Generators::FilterGenerator< T, Predicate >:



### Public Member Functions

- `template<typename P = Predicate>`  
**FilterGenerator** (P &&pred, [GeneratorWrapper](#)< T > &&generator)

- T const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.91.1 Member Function Documentation

#### 5.91.1.1 next()

```
template<typename T , typename Predicate >  
bool Catch::Generators::FilterGenerator< T, Predicate >::next ( ) [inline], [override], [virtual]
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.92 Catch::TestSpec::FilterMatch Struct Reference

### Public Attributes

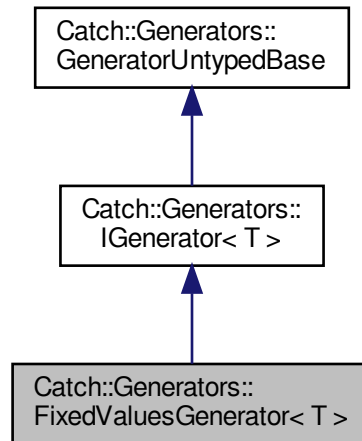
- std::string **name**
- std::vector< [TestCaseHandle](#) const \* > **tests**

The documentation for this struct was generated from the following file:

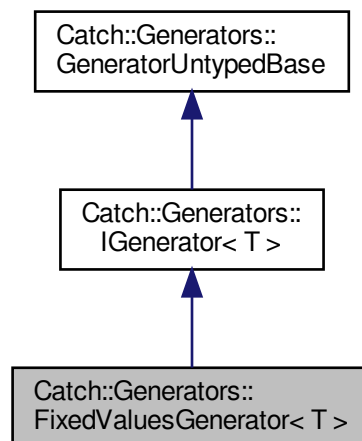
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.93 Catch::Generators::FixedValuesGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::FixedValuesGenerator< T >:



Collaboration diagram for Catch::Generators::FixedValuesGenerator< T >:



### Public Member Functions

- **FixedValuesGenerator** (std::initializer\_list< T > values)
- T const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.93.1 Member Function Documentation

#### 5.93.1.1 next()

```
template<typename T >  
bool Catch::Generators::FixedValuesGenerator< T >::next ( ) [inline], [override], [virtual]
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

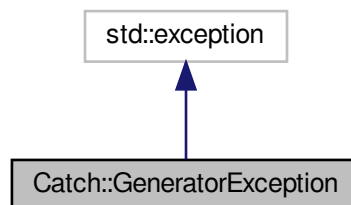
Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

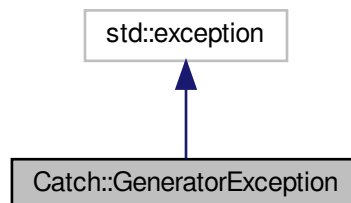
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.94 Catch::GeneratorException Class Reference

Inheritance diagram for Catch::GeneratorException:



Collaboration diagram for Catch::GeneratorException:





## Public Member Functions

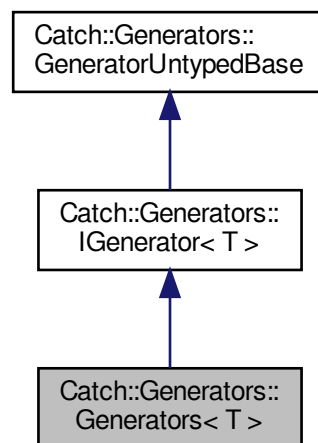
- **GeneratorException** (const char \*msg)
- const char \* **what** () const noexcept override final

The documentation for this class was generated from the following file:

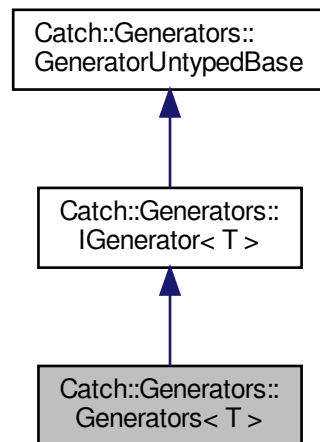
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.95 Catch::Generators::Generators< T > Class Template Reference

Inheritance diagram for Catch::Generators::Generators< T >:



Collaboration diagram for `Catch::Generators::Generators< T >`:



## Public Member Functions

- `template<typename... Gs>`  
**Generators** (Gs &&... moreGenerators)
- `T const & get ()` const override
- `bool next ()` override

## Additional Inherited Members

### 5.95.1 Member Function Documentation

#### 5.95.1.1 next()

```

template<typename T >
bool Catch::Generators::Generators< T >::next ( ) [inline], [override], [virtual]

```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

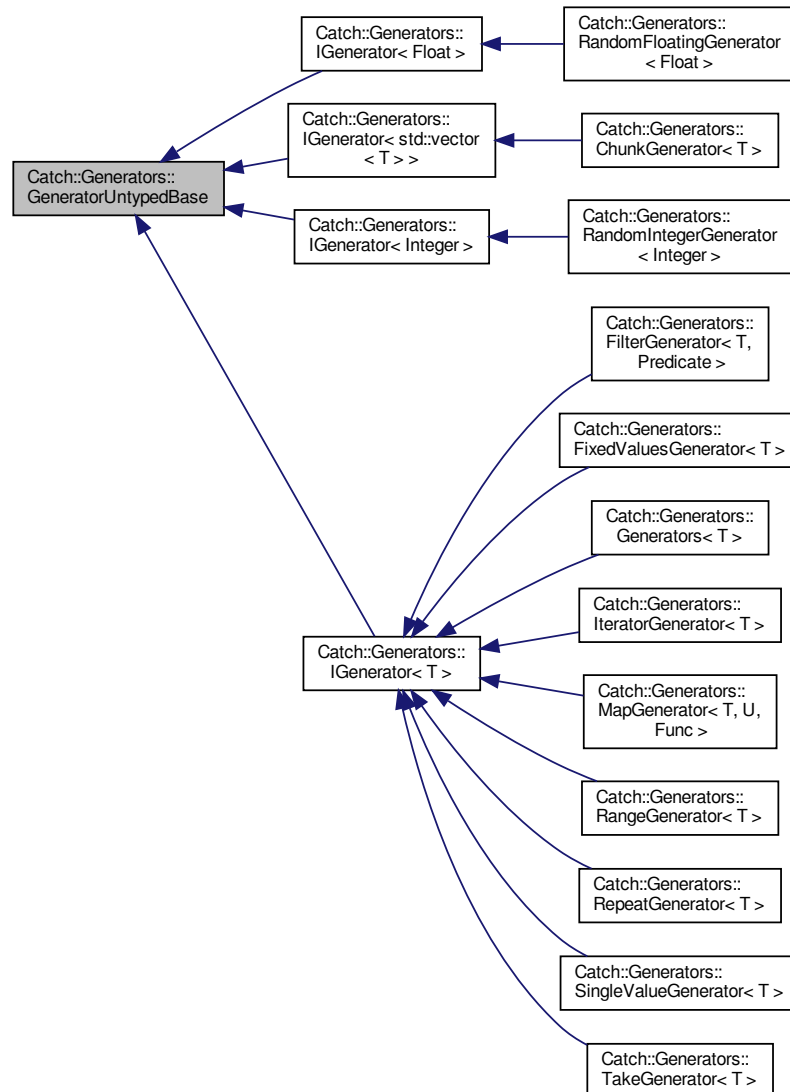
Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.96 Catch::Generators::GeneratorUntypedBase Class Reference

Inheritance diagram for Catch::Generators::GeneratorUntypedBase:



### Public Member Functions

- **GeneratorUntypedBase** ([GeneratorUntypedBase](#) const &)=default
- **GeneratorUntypedBase** & **operator=** ([GeneratorUntypedBase](#) const &)=default
- bool [countedNext](#) ()
- std::size\_t [currentElementIndex](#) () const
- [StringRef](#) [currentElementAsString](#) () const

#### 5.96.1 Member Function Documentation

### 5.96.1.1 countedNext()

```
bool Catch::Generators::GeneratorUntypedBase::countedNext ( )
```

Attempts to move the generator to the next element

Serves as a non-virtual interface to `next`, so that the top level interface can provide sanity checking and shared features.

As with `next`, returns true iff the move succeeded and the generator has new valid element to provide.

### 5.96.1.2 currentElementAsString()

```
StringRef Catch::Generators::GeneratorUntypedBase::currentElementAsString ( ) const
```

Returns generator's current element as user-friendly string.

By default returns string equivalent to calling `Catch::Detail::stringify` on the current element, but generators can customize their implementation as needed.

Not thread-safe due to internal caching.

The returned ref is valid only until the generator instance is destructed, or it moves onto the next element, whichever comes first.

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.97 Catch::Generators::GeneratorWrapper< T > Class Template Reference

### Public Member Functions

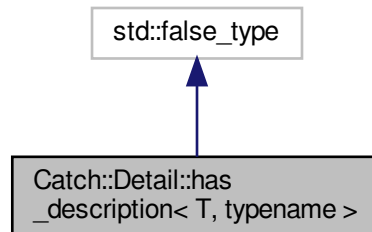
- [GeneratorWrapper](#) ([IGenerator](#)< T > \*generator)  
*Takes ownership of the passed pointer.*
- **GeneratorWrapper** ([GeneratorPtr](#)< T > generator)
- T const & **get** () const
- bool **next** ()

The documentation for this class was generated from the following file:

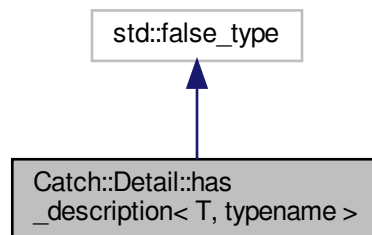
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.98 Catch::Detail::has\_description< T, typename > Struct Template Reference

Inheritance diagram for Catch::Detail::has\_description< T, typename >:



Collaboration diagram for Catch::Detail::has\_description< T, typename >:

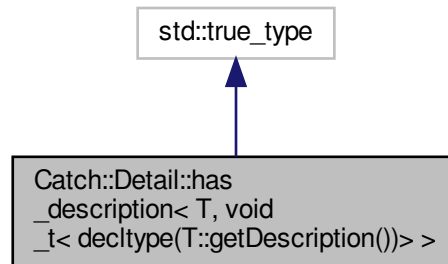


The documentation for this struct was generated from the following file:

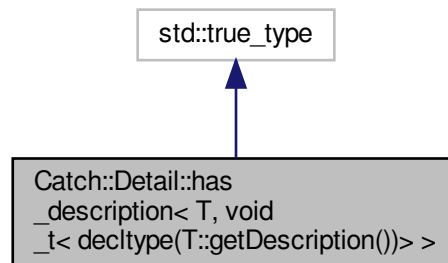
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.99 Catch::Detail::has\_description< T, void\_t< decltype(T::getDescription())> > Struct Template Reference

Inheritance diagram for Catch::Detail::has\_description< T, void\_t< decltype(T::getDescription())> >:



Collaboration diagram for Catch::Detail::has\_description< T, void\_t< decltype(T::getDescription())> >:

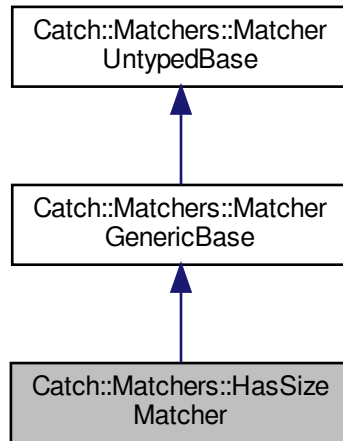


The documentation for this struct was generated from the following file:

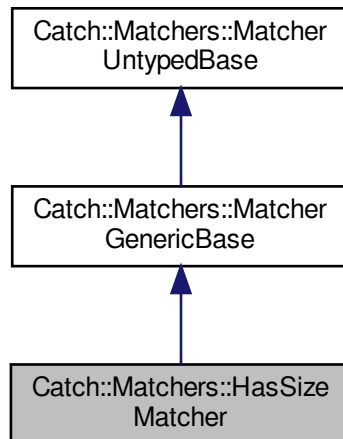
- `include/catch2/catch_amalgamated.hpp`

## 5.100 Catch::Matchers::HasSizeMatcher Class Reference

Inheritance diagram for Catch::Matchers::HasSizeMatcher:



Collaboration diagram for Catch::Matchers::HasSizeMatcher:



### Public Member Functions

- **HasSizeMatcher** (std::size\_t target\_size)
- template<typename RangeLike >  
bool **match** (RangeLike &&rng) const
- std::string **describe** () const override

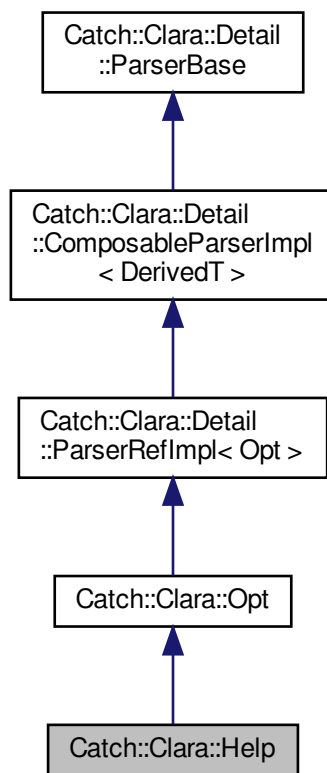
## Additional Inherited Members

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

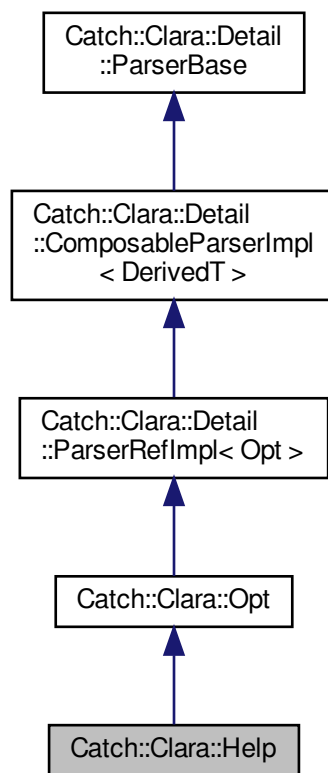
### 5.101 Catch::Clara::Help Struct Reference

Inheritance diagram for Catch::Clara::Help:





Collaboration diagram for Catch::Clara::Help:



### Public Member Functions

- **Help** (bool &showHelpFlag)

### Additional Inherited Members

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.102 Catch::Clara::Detail::HelpColumns Struct Reference

### Public Attributes

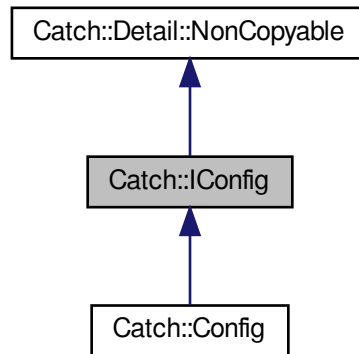
- std::string **left**
- std::string **right**

The documentation for this struct was generated from the following file:

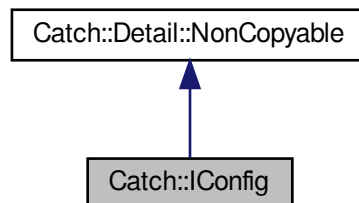
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.103 Catch::IConfig Class Reference

Inheritance diagram for Catch::IConfig:



Collaboration diagram for Catch::IConfig:



### Public Member Functions

- virtual bool **allowThrows** () const =0
- virtual [StringRef](#) **name** () const =0
- virtual bool **includeSuccessfulResults** () const =0
- virtual bool **shouldDebugBreak** () const =0
- virtual bool **warnAboutMissingAssertions** () const =0
- virtual bool **warnAboutUnmatchedTestSpecs** () const =0
- virtual bool **zeroTestsCountAsSuccess** () const =0
- virtual int **abortAfter** () const =0
- virtual bool **showInvisibles** () const =0
- virtual ShowDurations **showDurations** () const =0
- virtual double **minDuration** () const =0
- virtual [TestSpec](#) const & **testSpec** () const =0

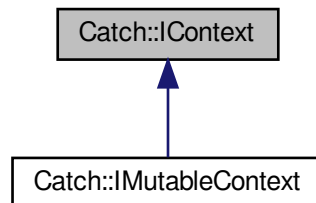
- virtual bool **hasTestFilters** () const =0
- virtual std::vector< std::string > const & **getTestsOrTags** () const =0
- virtual TestRunOrder **runOrder** () const =0
- virtual uint32\_t **rngSeed** () const =0
- virtual unsigned int **shardCount** () const =0
- virtual unsigned int **shardIndex** () const =0
- virtual ColourMode **defaultColourMode** () const =0
- virtual std::vector< std::string > const & **getSectionsToRun** () const =0
- virtual Verbosity **verbosity** () const =0
- virtual bool **skipBenchmarks** () const =0
- virtual bool **benchmarkNoAnalysis** () const =0
- virtual unsigned int **benchmarkSamples** () const =0
- virtual double **benchmarkConfidenceInterval** () const =0
- virtual unsigned int **benchmarkResamples** () const =0
- virtual std::chrono::milliseconds **benchmarkWarmupTime** () const =0

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.104 Catch::IContext Class Reference

Inheritance diagram for Catch::IContext:



### Public Member Functions

- virtual IResultCapture \* **getResultCapture** ()=0
- virtual IConfig const \* **getConfig** () const =0

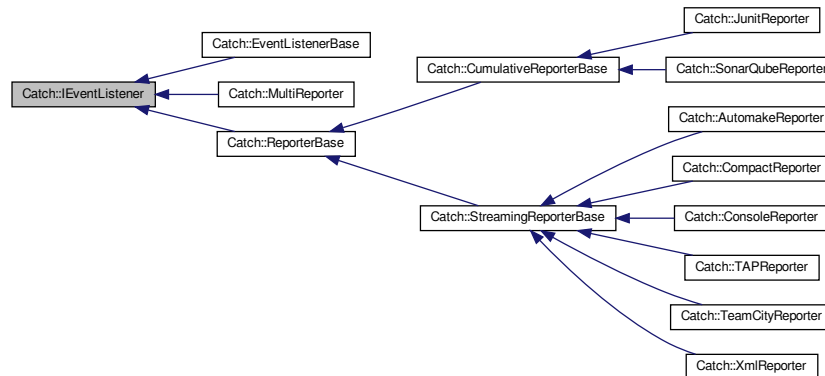
The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

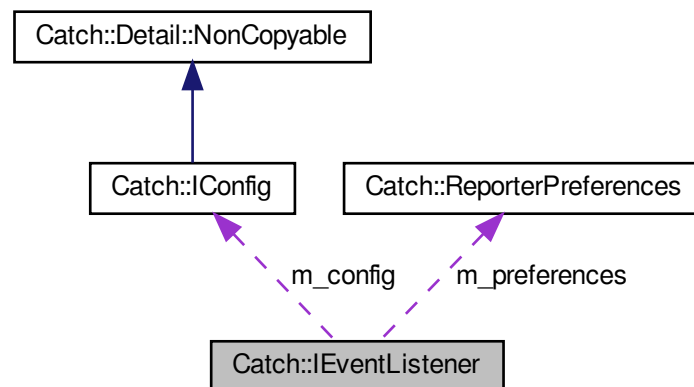
## 5.105 Catch::EventListener Class Reference

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for Catch::EventListener:



Collaboration diagram for Catch::EventListener:



### Public Member Functions

- **EventListener** ([IConfig](#) const \*config)
- [ReporterPreferences](#) const & **getPreferences** () const
- virtual void **noMatchingTestCases** ([StringRef](#) unmatchedSpec)=0  
*Called when no test cases match provided test spec.*
- virtual void **reportInvalidTestSpec** ([StringRef](#) invalidArgument)=0  
*Called for all invalid test specs from the cli.*
- virtual void **testRunStarting** ([TestRunInfo](#) const &testRunInfo)=0

- virtual void [testCaseStarting](#) ([TestCaseInfo](#) const &testInfo)=0  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- virtual void [testCasePartialStarting](#) ([TestCaseInfo](#) const &testInfo, uint64\_t partNumber)=0  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- virtual void [sectionStarting](#) ([SectionInfo](#) const &sectionInfo)=0  
*Called when a SECTION is being entered. Not called for skipped sections.*
- virtual void [benchmarkPreparing](#) ([StringRef](#) benchmarkName)=0  
*Called when user-code is being probed before the actual benchmark runs.*
- virtual void [benchmarkStarting](#) ([BenchmarkInfo](#) const &benchmarkInfo)=0  
*Called after probe but before the user-code is being benchmarked.*
- virtual void [benchmarkEnded](#) ([BenchmarkStats](#)<> const &benchmarkStats)=0  
*Called with the benchmark results if benchmark successfully finishes.*
- virtual void [benchmarkFailed](#) ([StringRef](#) benchmarkName)=0  
*Called if running the benchmarks fails for any reason.*
- virtual void [assertionStarting](#) ([AssertionInfo](#) const &assertionInfo)=0  
*Called before assertion success/failure is evaluated.*
- virtual void [assertionEnded](#) ([AssertionStats](#) const &assertionStats)=0  
*Called after assertion was fully evaluated.*
- virtual void [sectionEnded](#) ([SectionStats](#) const &sectionStats)=0  
*Called after a SECTION has finished running.*
- virtual void [testCasePartialEnded](#) ([TestCaseStats](#) const &testCaseStats, uint64\_t partNumber)=0  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- virtual void [testCaseEnded](#) ([TestCaseStats](#) const &testCaseStats)=0  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- virtual void [testRunEnded](#) ([TestRunStats](#) const &testRunStats)=0
- virtual void [skipTest](#) ([TestCaseInfo](#) const &testInfo)=0  
*Called with test cases that are skipped due to the test run aborting.*
- virtual void [fatalErrorEncountered](#) ([StringRef](#) error)=0  
*Called if a fatal error (signal/structured exception) occurred.*
- virtual void [listReporters](#) (std::vector< [ReporterDescription](#) > const &descriptions)=0  
*Writes out information about provided reporters using reporter-specific format.*
- virtual void [listListeners](#) (std::vector< [ListenerDescription](#) > const &descriptions)=0  
*Writes out the provided listeners descriptions using reporter-specific format.*
- virtual void [listTests](#) (std::vector< [TestCaseHandle](#) > const &tests)=0  
*Writes out information about provided tests using reporter-specific format.*
- virtual void [listTags](#) (std::vector< [TagInfo](#) > const &tags)=0  
*Writes out information about the provided tags using reporter-specific format.*

## Protected Attributes

- [ReporterPreferences](#) m\_preferences  
*Derived classes can set up their preferences here.*
- [IConfig](#) const \* m\_config  
*The test run's config as filled in from CLI and defaults.*

### 5.105.1 Detailed Description

The common base for all reporters and event listeners

Implementing classes must also implement:

```
//! User-friendly description of the reporter/listener type
static std::string getDescription()
```

Generally shouldn't be derived from by users of Catch2 directly, instead they should derive from one of the utility bases that derive from this class.

### 5.105.2 Member Function Documentation

#### 5.105.2.1 testRunEnded()

```
virtual void Catch::IEventListener::testRunEnded (
    TestRunStats const & testRunStats ) [pure virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Implemented in [Catch::XmlReporter](#), [Catch::MultiReporter](#), [Catch::EventListenerBase](#), [Catch::CumulativeReporterBase](#), [Catch::TeamCityReporter](#), [Catch::TAPReporter](#), [Catch::ConsoleReporter](#), [Catch::CompactReporter](#), and [Catch::StreamingReporterBase](#).

#### 5.105.2.2 testRunStarting()

```
virtual void Catch::IEventListener::testRunStarting (
    TestRunInfo const & testRunInfo ) [pure virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Implemented in [Catch::SonarQubeReporter](#), [Catch::MultiReporter](#), [Catch::EventListenerBase](#), [Catch::XmlReporter](#), [Catch::TAPReporter](#), [Catch::JUnitReporter](#), [Catch::TeamCityReporter](#), [Catch::ConsoleReporter](#), [Catch::StreamingReporterBase](#), [Catch::CompactReporter](#), and [Catch::CumulativeReporterBase](#).

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.106 Catch::IExceptionTranslator Class Reference

### Public Member Functions

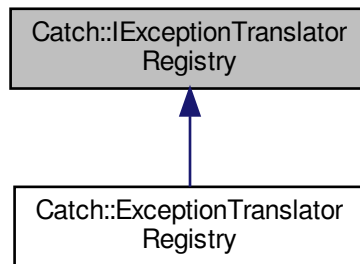
- virtual std::string **translate** (ExceptionTranslators::const\_iterator it, ExceptionTranslators::const\_iterator it↔ End) const =0

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.107 Catch::IExceptionTranslatorRegistry Class Reference

Inheritance diagram for Catch::IExceptionTranslatorRegistry:



### Public Member Functions

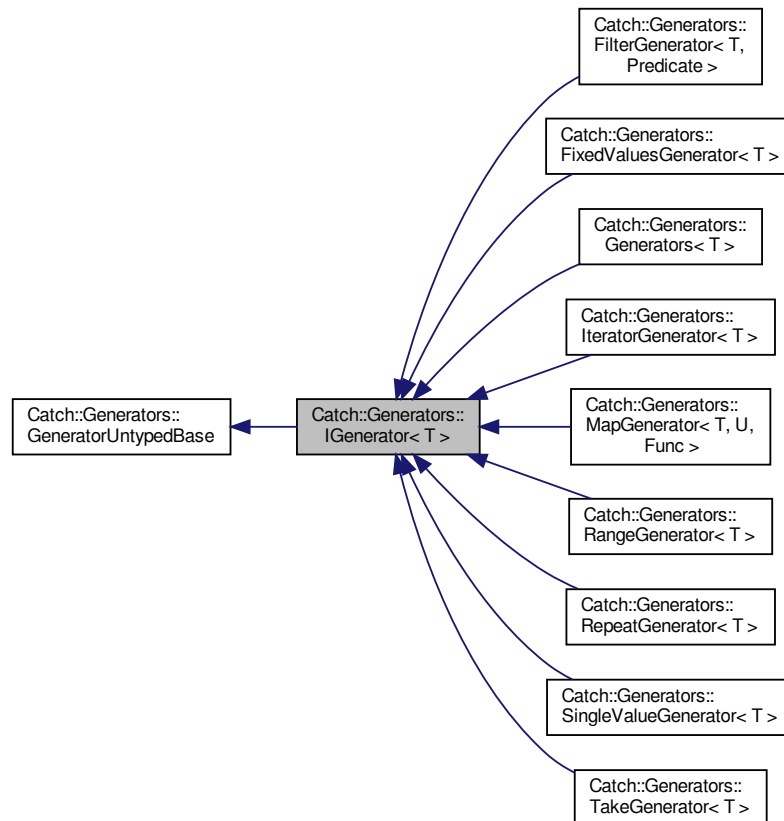
- virtual std::string **translateActiveException** () const =0

The documentation for this class was generated from the following file:

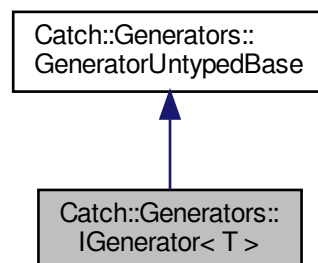
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.108 Catch::Generators::IGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::IGenerator< T >:



Collaboration diagram for Catch::Generators::IGenerator< T >:



### Public Types

- using `type` = T



## Public Member Functions

- **IGenerator** ([IGenerator](#) const &)=default
- **IGenerator** & **operator=** ([IGenerator](#) const &)=default
- virtual T const & **get** () const =0

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.109 Catch::IGeneratorTracker Class Reference

### Public Member Functions

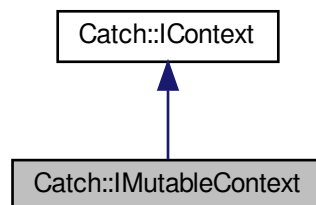
- virtual auto **hasGenerator** () const -> bool=0
- virtual auto **getGenerator** () const -> [Generators::GeneratorBasePtr](#) const &=0
- virtual void **setGenerator** ([Generators::GeneratorBasePtr](#) &&generator)=0

The documentation for this class was generated from the following file:

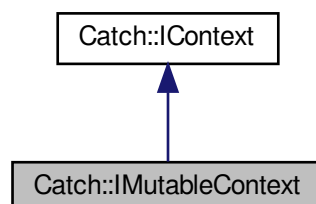
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.110 Catch::IMutableContext Class Reference

Inheritance diagram for Catch::IMutableContext:



Collaboration diagram for Catch::IMutableContext:



## Public Member Functions

- virtual void **setResultCapture** ([IResultCapture](#) \*resultCapture)=0
- virtual void **setConfig** ([IConfig](#) const \*config)=0

## Friends

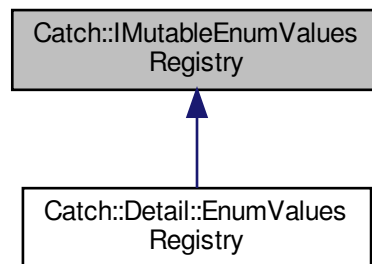
- [IMutableContext](#) & **getCurrentMutableContext** ()
- void **cleanUpContext** ()

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.111 Catch::IMutableEnumValuesRegistry Class Reference

Inheritance diagram for Catch::IMutableEnumValuesRegistry:



## Public Member Functions

- virtual [Detail::EnumInfo](#) const & **registerEnum** ([StringRef](#) enumName, [StringRef](#) allEnums, std::vector< int > const &values)=0
- template<typename E >  
[Detail::EnumInfo](#) const & **registerEnum** ([StringRef](#) enumName, [StringRef](#) allEnums, std::initializer\_list< E > values)

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.112 Catch::IMutableRegistryHub Class Reference

### Public Member Functions

- virtual void **registerReporter** (std::string const &name, [IReporterFactoryPtr](#) factory)=0
- virtual void **registerListener** ([Detail::unique\\_ptr< EventListenerFactory >](#) factory)=0
- virtual void **registerTest** ([Detail::unique\\_ptr< TestCaseInfo >](#) &&testInfo, [Detail::unique\\_ptr< ITestInvoker >](#) &&invoker)=0
- virtual void **registerTranslator** ([Detail::unique\\_ptr< IExceptionTranslator >](#) &&translator)=0
- virtual void **registerTagAlias** (std::string const &alias, std::string const &tag, [SourceLineInfo](#) const &lineInfo)=0
- virtual void **registerStartupException** () noexcept=0
- virtual [IMutableEnumValuesRegistry](#) & **getMutableEnumValuesRegistry** ()=0

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.113 Catch::IRegistryHub Class Reference

### Public Member Functions

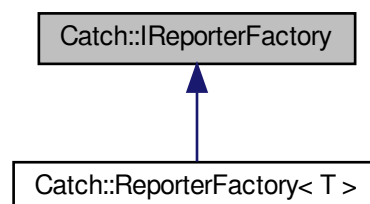
- virtual [IReporterRegistry](#) const & **getReporterRegistry** () const =0
- virtual [ITestCaseRegistry](#) const & **getTestCaseRegistry** () const =0
- virtual [ITagAliasRegistry](#) const & **getTagAliasRegistry** () const =0
- virtual [IExceptionTranslatorRegistry](#) const & **getExceptionTranslatorRegistry** () const =0
- virtual [StartupExceptionRegistry](#) const & **getStartupExceptionRegistry** () const =0

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.114 Catch::IReporterFactory Class Reference

Inheritance diagram for Catch::IReporterFactory:



## Public Member Functions

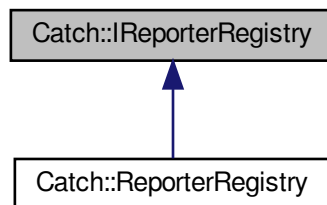
- virtual [IEventListenerPtr](#) **create** ([ReporterConfig](#) &&config) const =0
- virtual std::string **getDescription** () const =0

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.115 Catch::IReporterRegistry Class Reference

Inheritance diagram for Catch::IReporterRegistry:



## Public Types

- using **FactoryMap** = std::map< std::string, [IReporterFactoryPtr](#), [Detail::CaseInsensitiveLess](#) >
- using **Listeners** = std::vector< [Detail::unique\\_ptr](#)< [EventListenerFactory](#) > >

## Public Member Functions

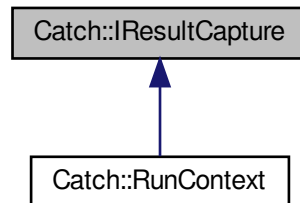
- virtual [IEventListenerPtr](#) **create** (std::string const &name, [ReporterConfig](#) &&config) const =0
- virtual FactoryMap const & **getFactories** () const =0
- virtual Listeners const & **getListeners** () const =0

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.116 Catch::IResultCapture Class Reference

Inheritance diagram for Catch::IResultCapture:



### Public Member Functions

- virtual bool **sectionStarted** ([SectionInfo](#) const &sectionInfo, [Counts](#) &assertions)=0
- virtual void **sectionEnded** ([SectionEndInfo](#) const &endInfo)=0
- virtual void **sectionEndedEarly** ([SectionEndInfo](#) const &endInfo)=0
- virtual auto **acquireGeneratorTracker** ([StringRef](#) generatorName, [SourceLineInfo](#) const &lineInfo) -> [IGeneratorTracker](#) &=0
- virtual void **benchmarkPreparing** ([StringRef](#) name)=0
- virtual void **benchmarkStarting** ([BenchmarkInfo](#) const &info)=0
- virtual void **benchmarkEnded** ([BenchmarkStats](#)<> const &stats)=0
- virtual void **benchmarkFailed** ([StringRef](#) error)=0
- virtual void **pushScopedMessage** ([MessageInfo](#) const &message)=0
- virtual void **popScopedMessage** ([MessageInfo](#) const &message)=0
- virtual void **emplaceUnscopedMessage** ([MessageBuilder](#) const &builder)=0
- virtual void **handleFatalErrorCondition** ([StringRef](#) message)=0
- virtual void **handleExpr** ([AssertionInfo](#) const &info, [ITransientExpression](#) const &expr, [AssertionReaction](#) &reaction)=0
- virtual void **handleMessage** ([AssertionInfo](#) const &info, [ResultWas::OfType](#) resultType, [StringRef](#) message, [AssertionReaction](#) &reaction)=0
- virtual void **handleUnexpectedExceptionNotThrown** ([AssertionInfo](#) const &info, [AssertionReaction](#) &reaction)=0
- virtual void **handleUnexpectedInflightException** ([AssertionInfo](#) const &info, std::string const &message, [AssertionReaction](#) &reaction)=0
- virtual void **handleIncomplete** ([AssertionInfo](#) const &info)=0
- virtual void **handleNonExpr** ([AssertionInfo](#) const &info, [ResultWas::OfType](#) resultType, [AssertionReaction](#) &reaction)=0
- virtual bool **lastAssertionPassed** ()=0
- virtual void **assertionPassed** ()=0
- virtual std::string **getCurrentTestName** () const =0
- virtual const [AssertionResult](#) \* **getLastResult** () const =0
- virtual void **exceptionEarlyReported** ()=0

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

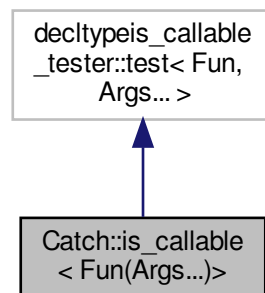
### 5.117 `Catch::is_callable< T >` Struct Template Reference

The documentation for this struct was generated from the following file:

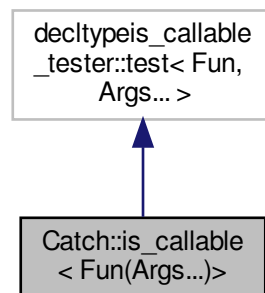
- `include/catch2/catch_amalgamated.hpp`

### 5.118 `Catch::is_callable< Fun(Args...)>` Struct Template Reference

Inheritance diagram for `Catch::is_callable< Fun(Args...)>`:



Collaboration diagram for `Catch::is_callable< Fun(Args...)>`:



The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.119 Catch::is\_callable\_tester Struct Reference

### Static Public Member Functions

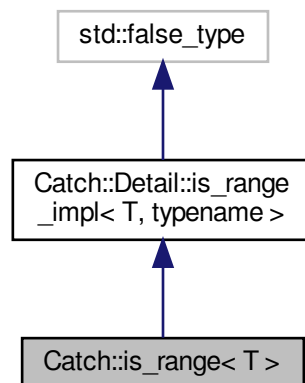
- `template<typename Fun , typename... Args>  
static true\_given< decltype(std::declval< Fun >)(std::declval< Args >)...>> test (int)`
- `template<typename... >  
static std::false_type test (...)`

The documentation for this struct was generated from the following file:

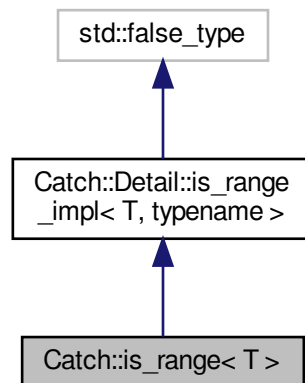
- `include/catch2/catch\_amalgamated.hpp`

## 5.120 Catch::is\_range< T > Struct Template Reference

Inheritance diagram for Catch::is\_range< T >:



Collaboration diagram for `Catch::is_range< T >`:

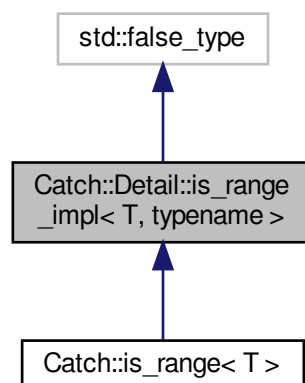


The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

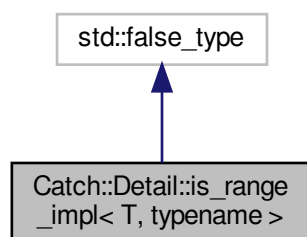
## 5.121 `Catch::Detail::is_range_impl< T, typename >` Struct Template Reference

Inheritance diagram for `Catch::Detail::is_range_impl< T, typename >`:





Collaboration diagram for `Catch::Detail::is_range_impl< T, typename >`:

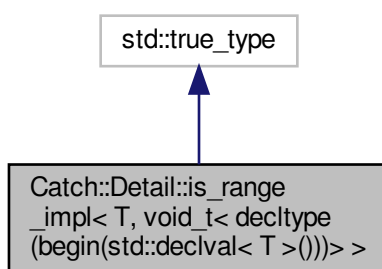


The documentation for this struct was generated from the following file:

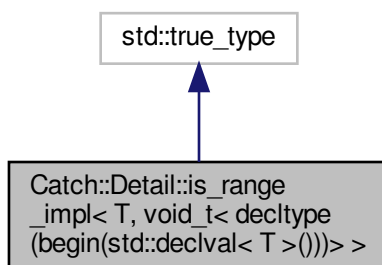
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.122 **Catch::Detail::is\_range\_impl**< T, void\_t< decltype(begin(std::declval< T >()))>> > Struct Template Reference

Inheritance diagram for `Catch::Detail::is_range_impl< T, void_t< decltype(begin(std::declval< T >()))>> >`:



Collaboration diagram for `Catch::Detail::is_range_impl< T, void_t< decltype(begin(std::declval< T >()))>> >`:

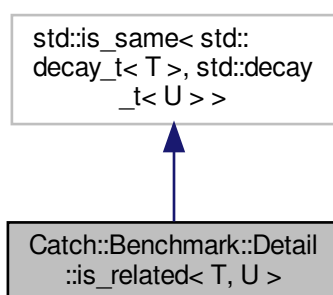


The documentation for this struct was generated from the following file:

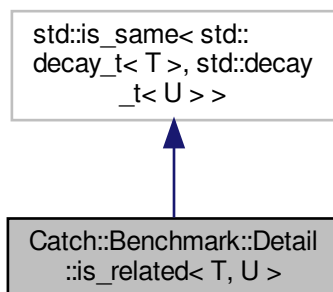
- `include/catch2/catch_amalgamated.hpp`

### 5.123 `Catch::Benchmark::Detail::is_related< T, U >` Struct Template Reference

Inheritance diagram for `Catch::Benchmark::Detail::is_related< T, U >`:



Collaboration diagram for Catch::Benchmark::Detail::is\_related< T, U >:

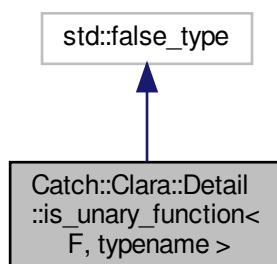


The documentation for this struct was generated from the following file:

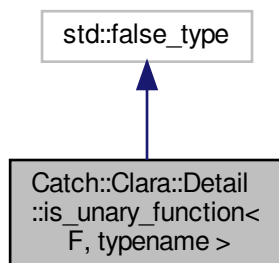
- `include/catch2/catch_amalgamated.hpp`

## 5.124 Catch::Clara::Detail::is\_unary\_function< F, typename > Struct Template Reference

Inheritance diagram for Catch::Clara::Detail::is\_unary\_function< F, typename >:



Collaboration diagram for `Catch::Clara::Detail::is_unary_function< F, typename >`:

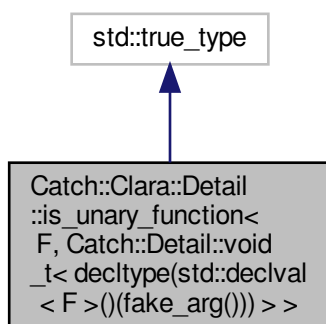


The documentation for this struct was generated from the following file:

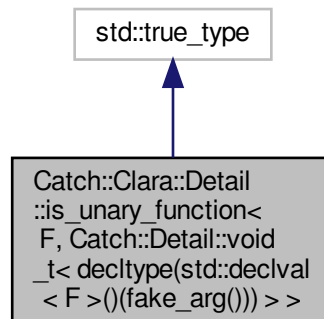
- `include/catch2/catch_amalgamated.hpp`

## 5.125 `Catch::Clara::Detail::is_unary_function< F, Catch::Detail::void_t< decltype(std::declval< F >())(fake_arg()) > >` Struct Template Reference

Inheritance diagram for `Catch::Clara::Detail::is_unary_function< F, Catch::Detail::void_t< decltype(std::declval< F >())(fake_arg()) > >`:



Collaboration diagram for Catch::Clara::Detail::is\_unary\_function< F, Catch::Detail::void\_t< decltype(std::declval< F >()(fake\_arg())) > >:

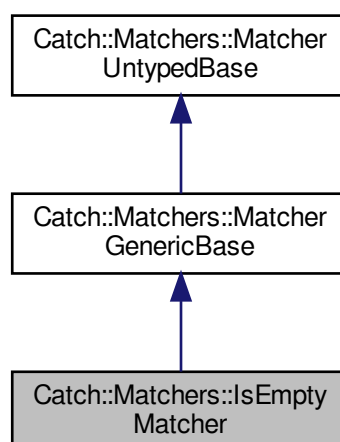


The documentation for this struct was generated from the following file:

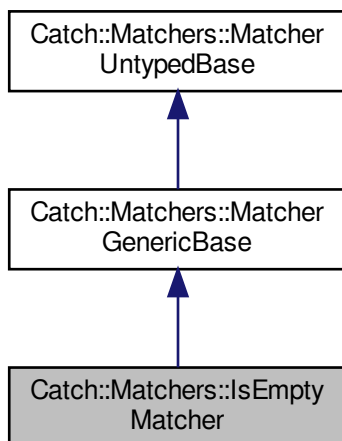
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.126 Catch::Matchers::IsEmptyMatcher Class Reference

Inheritance diagram for Catch::Matchers::IsEmptyMatcher:



Collaboration diagram for `Catch::Matchers::IsEmptyMatcher`:



## Public Member Functions

- `template<typename RangeLike >`  
`bool match (RangeLike &&rng) const`
- `std::string describe () const` override

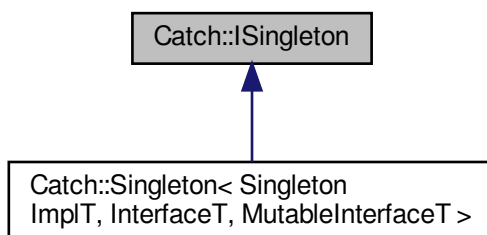
## Additional Inherited Members

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.127 Catch::ISingleton Struct Reference

Inheritance diagram for `Catch::ISingleton`:



The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.128 Catch::Detail::IsStreamInsertable< T > Class Template Reference

### Static Public Attributes

- static const bool **value** = decltype(test<std::ostream, const T&>(0))::value

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.129 Catch::IStream Class Reference

### Public Member Functions

- virtual std::ostream & **stream** ()=0
- virtual bool [isConsole](#) () const

### 5.129.1 Member Function Documentation

#### 5.129.1.1 isConsole()

```
virtual bool Catch::IStream::isConsole ( ) const [inline], [virtual]
```

Best guess on whether the instance is writing to a console (e.g. via stdout/stderr)

This is useful for e.g. Win32 colour support, because the Win32 API manipulates console directly, unlike POSIX escape codes, that can be written anywhere.

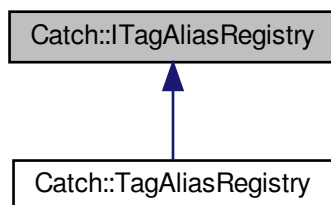
Due to variety of ways to change where the stdout/stderr is *actually* being written, users should always assume that the answer might be wrong.

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.130 Catch::ITagAliasRegistry Class Reference

Inheritance diagram for Catch::ITagAliasRegistry:



### Public Member Functions

- virtual [TagAlias](#) const \* **find** (std::string const &alias) const =0
- virtual std::string **expandAliases** (std::string const &unexpandedTestSpec) const =0

### Static Public Member Functions

- static [ITagAliasRegistry](#) const & **get** ()

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.131 Catch::TextFlow::Columns::iterator Class Reference

### Public Types

- using **difference\_type** = std::ptrdiff\_t
- using **value\_type** = std::string
- using **pointer** = value\_type \*
- using **reference** = value\_type &
- using **iterator\_category** = std::forward\_iterator\_tag

### Public Member Functions

- **iterator** ([Columns](#) const &columns)
- auto **operator==** ([iterator](#) const &other) const -> bool
- auto **operator!=** ([iterator](#) const &other) const -> bool
- std::string **operator\*** () const
- [iterator](#) & **operator++** ()
- [iterator](#) **operator++** (int)

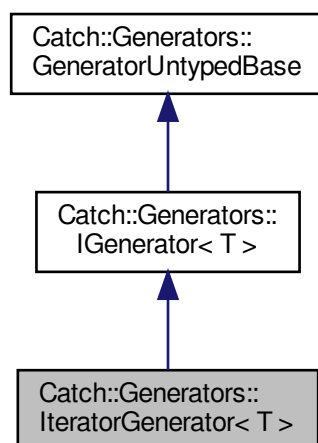
The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

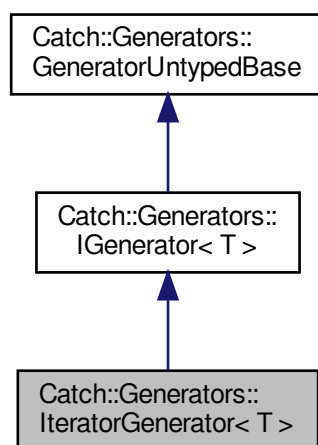


## 5.132 Catch::Generators::IteratorGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::IteratorGenerator< T >:



Collaboration diagram for Catch::Generators::IteratorGenerator< T >:



### Public Member Functions

- `template<typename InputIterator, typename InputSentinel >`  
**IteratorGenerator** (InputIterator first, InputSentinel last)
- `T const & get ()` const override
- `bool next ()` override

## Additional Inherited Members

### 5.132.1 Member Function Documentation

#### 5.132.1.1 next()

```
template<typename T >
bool Catch::Generators::IteratorGenerator< T >::next ( ) [inline], [override], [virtual]
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

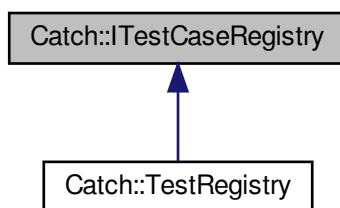
Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.133 Catch::ITestCaseRegistry Class Reference

Inheritance diagram for Catch::ITestCaseRegistry:



### Public Member Functions

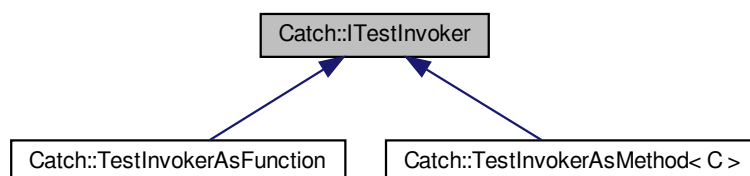
- virtual std::vector< [TestCaseInfo](#) \* > const & **getAllInfos** () const =0
- virtual std::vector< [TestCaseHandle](#) > const & **getAllTests** () const =0
- virtual std::vector< [TestCaseHandle](#) > const & **getAllTestsSorted** ([IConfig](#) const &config) const =0

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.134 Catch::ITestInvoker Class Reference

Inheritance diagram for Catch::ITestInvoker:



### Public Member Functions

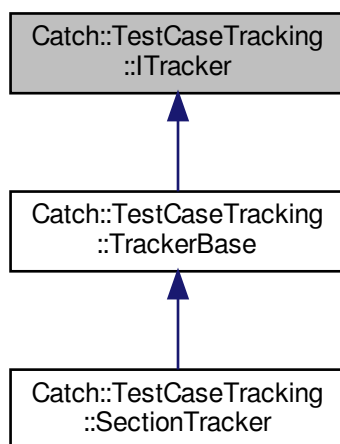
- virtual void **invoke** () const =0

The documentation for this class was generated from the following file:

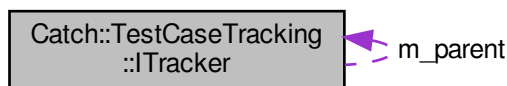
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.135 Catch::TestCaseTracking::ITracker Class Reference

Inheritance diagram for Catch::TestCaseTracking::ITracker:



Collaboration diagram for Catch::TestCaseTracking::ITracker:



## Public Member Functions

- **ITracker** ([NameAndLocation](#) const &nameAndLoc, [ITracker](#) \*parent)
- [NameAndLocation](#) const & **nameAndLocation** () const
- [ITracker](#) \* **parent** () const
- virtual bool [isComplete](#) () const =0  
*Returns true if tracker run to completion (successfully or not)*
- bool [isSuccessfullyCompleted](#) () const  
*Returns true if tracker run to completion succesfully.*
- bool [isOpen](#) () const  
*Returns true if tracker has started but hasn't been completed.*
- bool [hasStarted](#) () const  
*Returns true iff tracker has started.*
- virtual void **close** ()=0
- virtual void **fail** ()=0
- void **markAsNeedingAnotherRun** ()
- void [addChild](#) ([ITrackerPtr](#) &&child)  
*Register a nested ITracker.*
- [ITracker](#) \* [findChild](#) ([NameAndLocation](#) const &nameAndLocation)
- bool [hasChildren](#) () const  
*Have any children been added?*
- void [openChild](#) ()  
*Marks tracker as executing a child, doing se recursively up the tree.*
- virtual bool [isSectionTracker](#) () const
- virtual bool [isGeneratorTracker](#) () const

## Protected Types

- enum **CycleState** {  
  **NotStarted** , **Executing** , **ExecutingChildren** , **NeedsAnotherRun** ,  
  **CompletedSuccessfully** , **Failed** }

## Protected Attributes

- [ITracker](#) \* **m\_parent** = nullptr
- Children **m\_children**
- CycleState **m\_runState** = NotStarted

## 5.135.1 Member Function Documentation

### 5.135.1.1 findChild()

```
ITracker* Catch::TestCaseTracking::ITracker::findChild (
    NameAndLocation const & nameAndLocation )
```

Returns ptr to specific child if register with this tracker.

Returns nullptr if not found.

### 5.135.1.2 isGeneratorTracker()

```
virtual bool Catch::TestCaseTracking::ITracker::isGeneratorTracker ( ) const [virtual]
```

Returns true if the instance is a generator tracker

Subclasses should override to true if they are, replaces RTTI for internal debug checks.

### 5.135.1.3 isSectionTracker()

```
virtual bool Catch::TestCaseTracking::ITracker::isSectionTracker ( ) const [virtual]
```

Returns true if the instance is a section tracker

Subclasses should override to true if they are, replaces RTTI for internal debug checks.

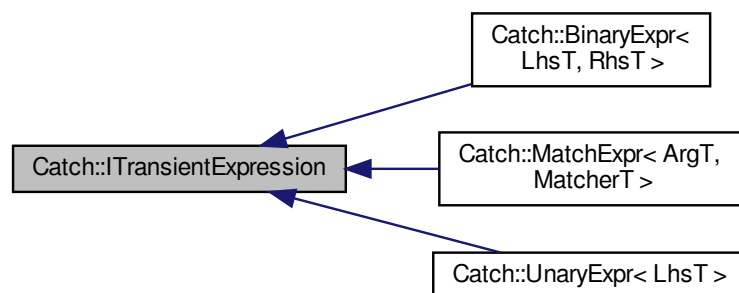
Reimplemented in [Catch::TestCaseTracking::SectionTracker](#).

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.136 Catch::ITransientExpression Class Reference

Inheritance diagram for Catch::ITransientExpression:



## Public Member Functions

- auto **isBinaryExpression** () const -> bool
- auto **getResult** () const -> bool
- virtual void **streamReconstructedExpression** (std::ostream &os) const =0
- **ITransientExpression** (bool isBinaryExpression, bool result)
- **ITransientExpression** (**ITransientExpression** const &)=default
- **ITransientExpression** & **operator=** (**ITransientExpression** const &)=default

## Friends

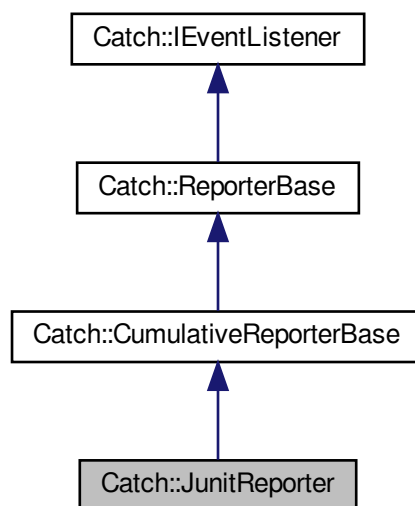
- std::ostream & **operator<<** (std::ostream &out, **ITransientExpression** const &expr)

The documentation for this class was generated from the following file:

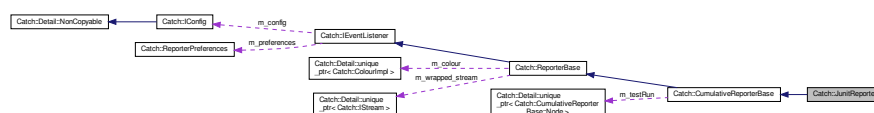
- include/catch2/catch\_amalgamated.hpp

## 5.137 Catch::JunitReporter Class Reference

Inheritance diagram for Catch::JunitReporter:



Collaboration diagram for Catch::JunitReporter:



## Public Member Functions

- **JUnitReporter** ([ReporterConfig](#) &&\_config)
- void [testRunStarting](#) ([TestRunInfo](#) const &runInfo) override
- void [testCaseStarting](#) ([TestCaseInfo](#) const &testCaseInfo) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [assertionEnded](#) ([AssertionStats](#) const &assertionStats) override  
*Called after assertion was fully evaluated.*
- void [testCaseEnded](#) ([TestCaseStats](#) const &testCaseStats) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testRunEndedCumulative](#) () override  
*Customization point: called after last test finishes (testRunEnded has been handled)*

## Static Public Member Functions

- static std::string [getDescription](#) ()

## Additional Inherited Members

### 5.137.1 Member Function Documentation

#### 5.137.1.1 testRunStarting()

```
void Catch::JUnitReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Reimplemented from [Catch::CumulativeReporterBase](#).

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.138 Catch::Clara::Detail::LambdaInvoker< ReturnType > Struct Template Reference

### Static Public Member Functions

- template<typename L, typename ArgType >  
static auto **invoke** (L const &lambda, ArgType const &arg) -> [ParserResult](#)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.139 Catch::Clara::Detail::LambdaInvoker< void > Struct Reference

### Static Public Member Functions

- `template<typename L , typename ArgType >`  
`static auto invoke (L const &lambda, ArgType const &arg) -> ParserResult`

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.140 Catch::LazyExpression Class Reference

### Public Member Functions

- **LazyExpression** (bool isNegated)
- **LazyExpression** ([LazyExpression](#) const &other)=default
- [LazyExpression](#) & **operator=** ([LazyExpression](#) const &)=delete
- **operator bool** () const

### Friends

- class **AssertionHandler**
- struct **AssertionStats**
- class **RunContext**
- `auto operator<< (std::ostream &os, LazyExpression const &lazyExpr) -> std::ostream &`

The documentation for this class was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.141 Catch::LeakDetector Struct Reference

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.142 LFMCMC< TData > Class Template Reference

Likelihood-Free Markov Chain Monte Carlo.

```
#include <lfmcmc-bones.hpp>
```



## Public Member Functions

- void **run** (std::vector< epiworld\_double > param\_init, size\_t n\_samples\_, epiworld\_double epsilon\_)
- **LFMCMC** (TData &observed\_data\_)
- void **set\_observed\_data** (TData &observed\_data\_)
- void **set\_proposal\_fun** (LFMCMCProposalFun< TData > fun)
- void **set\_simulation\_fun** (LFMCMCSimFun< TData > fun)
- void **set\_summary\_fun** (LFMCMCSummaryFun< TData > fun)
- void **set\_kernel\_fun** (LFMCMCKernelFun< TData > fun)
- const size\_t **get\_n\_samples** ()
- const size\_t **get\_n\_statistics** ()
- const size\_t **get\_n\_parameters** ()
- const epiworld\_double **get\_epsilon** ()
- const std::vector< epiworld\_double > & **get\_params\_now** ()
- const std::vector< epiworld\_double > & **get\_params\_prev** ()
- const std::vector< epiworld\_double > & **get\_params\_init** ()
- const std::vector< epiworld\_double > & **get\_statistics\_obs** ()
- const std::vector< epiworld\_double > & **get\_statistics\_hist** ()
- const std::vector< bool > & **get\_statistics\_accepted** ()
- const std::vector< epiworld\_double > & **get\_posterior\_if\_prob** ()
- const std::vector< epiworld\_double > & **get\_drawn\_prob** ()
- std::vector< TData > \* **get\_sampled\_data** ()
- void **set\_par\_names** (std::vector< std::string > names)
- void **set\_stats\_names** (std::vector< std::string > names)
- void **print** ()

## Random number generation

### Parameters

eng	
-----	--

- void **set\_rand\_engine** (std::mt19937 &eng)
- std::mt19937 \* **get\_rand\_engine** ()
- void **seed** (unsigned int s)
- void **set\_rand\_gamma** (epiworld\_double alpha, epiworld\_double beta)
- epiworld\_double **runif** ()
- epiworld\_double **rnorm** ()
- epiworld\_double **rgamma** ()
- epiworld\_double **runif** (epiworld\_double lb, epiworld\_double ub)
- epiworld\_double **rnorm** (epiworld\_double mean, epiworld\_double sd)
- epiworld\_double **rgamma** (epiworld\_double alpha, epiworld\_double beta)

## 5.142.1 Detailed Description

```
template<typename TData>
class LFMCMC< TData >
```

Likelihood-Free Markov Chain Monte Carlo.

### Template Parameters

<i>TData</i>	Type of data that is generated
--------------	--------------------------------

The documentation for this class was generated from the following files:

- include/epiworld/math/lfmcmc/lfmcmc-bones.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat.hpp

## 5.143 Catch::lineOfChars Struct Reference

### Public Member Functions

- constexpr **lineOfChars** (char c\_)

### Public Attributes

- char **c**

### Friends

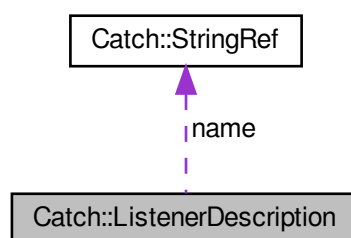
- std::ostream & **operator**<< (std::ostream &out, [lineOfChars](#) value)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.144 Catch::ListenerDescription Struct Reference

Collaboration diagram for Catch::ListenerDescription:



### Public Attributes

- [StringRef](#) **name**
- std::string **description**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.145 Catch::ListenerRegistrar< T > Class Template Reference

### Public Member Functions

- **ListenerRegistrar** ([StringRef](#) listenerName)

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.146 Catch::Detail::make\_void<... > Struct Template Reference

### Public Types

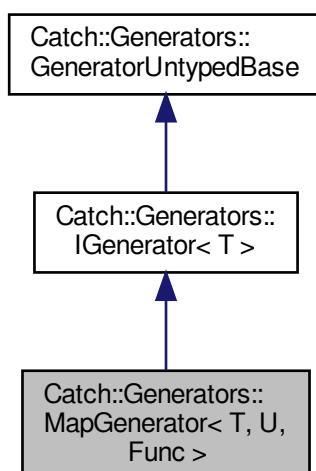
- using **type** = void

The documentation for this struct was generated from the following file:

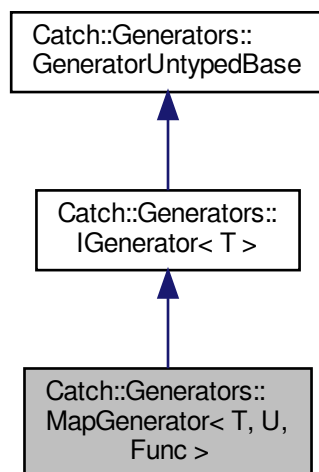
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.147 Catch::Generators::MapGenerator< T, U, Func > Class Template Reference

Inheritance diagram for Catch::Generators::MapGenerator< T, U, Func >:



Collaboration diagram for `Catch::Generators::MapGenerator< T, U, Func >`:



## Public Member Functions

- `template<typename F2 = Func>`  
**MapGenerator** (F2 &&function, [GeneratorWrapper](#)< U > &&generator)
- `T const & get ()` const override
- `bool next ()` override

## Additional Inherited Members

### 5.147.1 Member Function Documentation

#### 5.147.1.1 next()

```

template<typename T , typename U , typename Func >
bool Catch::Generators::MapGenerator< T, U, Func >::next ( ) [inline], [override], [virtual]

```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

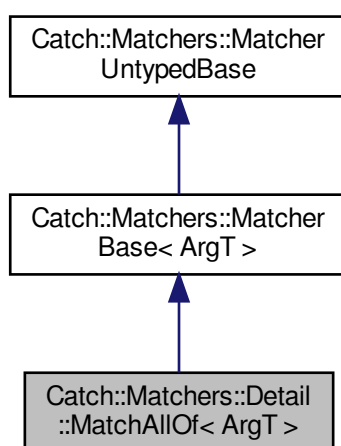
Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

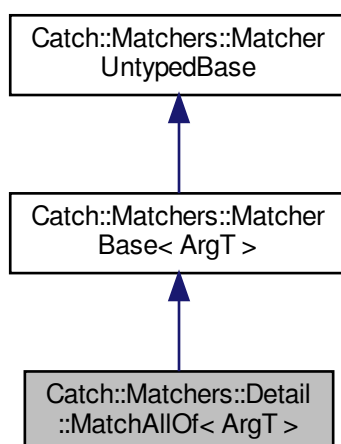
- `include/catch2/catch_amalgamated.hpp`

## 5.148 Catch::Matchers::Detail::MatchAllOf< ArgT > Class Template Reference

Inheritance diagram for Catch::Matchers::Detail::MatchAllOf< ArgT >:



Collaboration diagram for Catch::Matchers::Detail::MatchAllOf< ArgT >:



### Public Member Functions

- **MatchAllOf** ([MatchAllOf](#) const &)=delete

- [MatchAllOf](#) & **operator=** ([MatchAllOf](#) const &)=delete
- **MatchAllOf** ([MatchAllOf](#) &&)=default
- [MatchAllOf](#) & **operator=** ([MatchAllOf](#) &&)=default
- bool **match** (ArgT const &arg) const override
- std::string **describe** () const override

## Friends

- [MatchAllOf](#) **operator&&** ([MatchAllOf](#) &&lhs, [MatcherBase](#)< ArgT > const &rhs)
- [MatchAllOf](#) **operator&&** ([MatcherBase](#)< ArgT > const &lhs, [MatchAllOf](#) &&rhs)

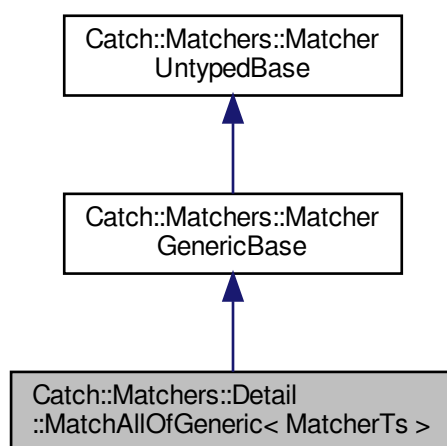
## Additional Inherited Members

The documentation for this class was generated from the following file:

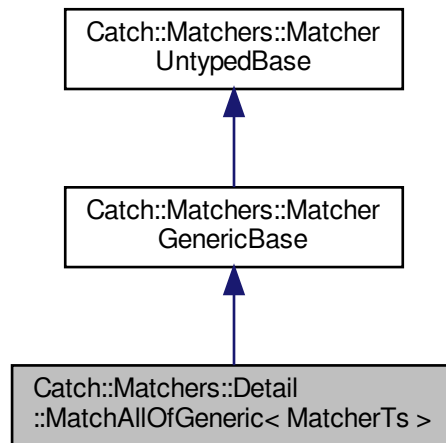
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.149 Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs > Class Template Reference

Inheritance diagram for Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs >:



Collaboration diagram for Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs >:



## Public Member Functions

- **MatchAllOfGeneric** ([MatchAllOfGeneric](#) const &)=delete
- **MatchAllOfGeneric** & **operator=** ([MatchAllOfGeneric](#) const &)=delete
- **MatchAllOfGeneric** ([MatchAllOfGeneric](#) &&)=default
- **MatchAllOfGeneric** & **operator=** ([MatchAllOfGeneric](#) &&)=default
- **MatchAllOfGeneric** (MatcherTs const &... matchers)
- **MatchAllOfGeneric** (std::array< void const \*, sizeof...(MatcherTs)> matchers)
- template<typename Arg >  
bool **match** (Arg &&arg) const
- std::string **describe** () const override

## Public Attributes

- std::array< void const \*, sizeof...(MatcherTs)> **m\_matchers**

## Friends

- template<typename... MatchersRHS>  
[MatchAllOfGeneric](#)< MatcherTs..., MatchersRHS... > **operator&&** ([MatchAllOfGeneric](#)< MatcherTs... > &&lhs, [MatchAllOfGeneric](#)< MatchersRHS... > &&rhs)  
*Avoids type nesting for GenericAllOf && GenericAllOf case.*
- template<typename MatcherRHS >  
std::enable\_if\_t< is\_matcher< MatcherRHS >::value, [MatchAllOfGeneric](#)< MatcherTs..., MatcherRHS > > **operator&&** ([MatchAllOfGeneric](#)< MatcherTs... > &&lhs, MatcherRHS const &rhs)  
*Avoids type nesting for GenericAllOf && some matcher case.*
- template<typename MatcherLHS >  
std::enable\_if\_t< is\_matcher< MatcherLHS >::value, [MatchAllOfGeneric](#)< MatcherLHS, MatcherTs... > > **operator&&** (MatcherLHS const &lhs, [MatchAllOfGeneric](#)< MatcherTs... > &&rhs)  
*Avoids type nesting for some matcher && GenericAllOf case.*

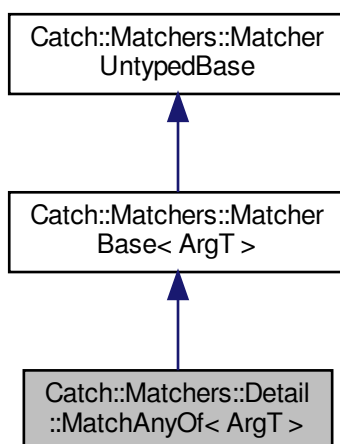
## Additional Inherited Members

The documentation for this class was generated from the following file:

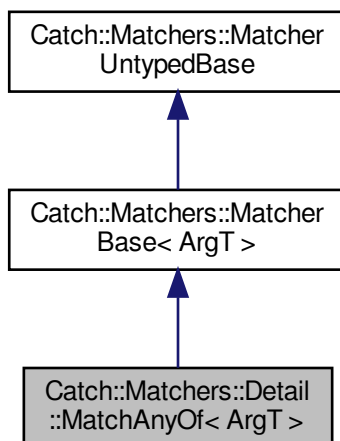
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.150 Catch::Matchers::Detail::MatchAnyOf< ArgT > Class Template Reference

Inheritance diagram for Catch::Matchers::Detail::MatchAnyOf< ArgT >:



Collaboration diagram for Catch::Matchers::Detail::MatchAnyOf< ArgT >:





## Public Member Functions

- **MatchAnyOf** ([MatchAnyOf](#) const &)=delete
- **MatchAnyOf** & **operator=** ([MatchAnyOf](#) const &)=delete
- **MatchAnyOf** ([MatchAnyOf](#) &&)=default
- **MatchAnyOf** & **operator=** ([MatchAnyOf](#) &&)=default
- bool **match** (ArgT const &arg) const override
- std::string **describe** () const override

## Friends

- **MatchAnyOf** **operator||** ([MatchAnyOf](#) &&lhs, [MatcherBase](#)< ArgT > const &rhs)
- **MatchAnyOf** **operator||** ([MatcherBase](#)< ArgT > const &lhs, [MatchAnyOf](#) &&rhs)

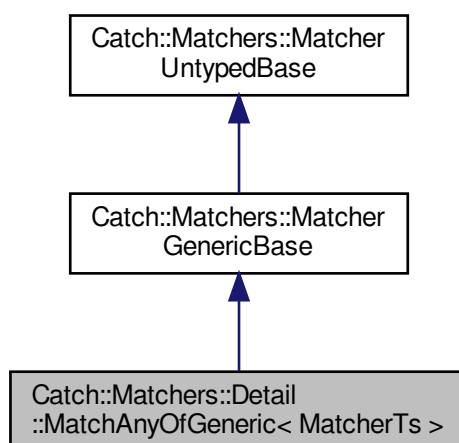
## Additional Inherited Members

The documentation for this class was generated from the following file:

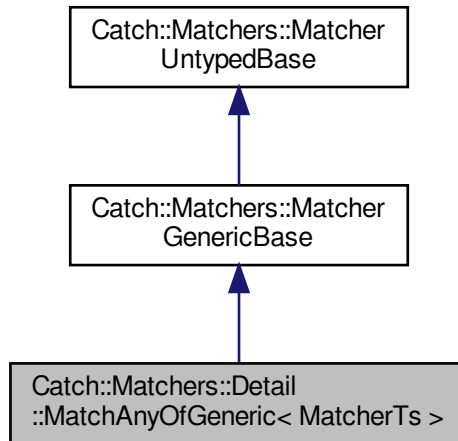
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.151 Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs > Class Template Reference

Inheritance diagram for Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs >:



Collaboration diagram for `Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs >`:



## Public Member Functions

- **MatchAnyOfGeneric** ([MatchAnyOfGeneric](#) const &)=delete
- **MatchAnyOfGeneric** & **operator=** ([MatchAnyOfGeneric](#) const &)=delete
- **MatchAnyOfGeneric** ([MatchAnyOfGeneric](#) &&)=default
- **MatchAnyOfGeneric** & **operator=** ([MatchAnyOfGeneric](#) &&)=default
- **MatchAnyOfGeneric** (MatcherTs const &... matchers)
- **MatchAnyOfGeneric** (std::array< void const \*, sizeof...(MatcherTs)> matchers)
- template<typename Arg >  
bool **match** (Arg &&arg) const
- std::string **describe** () const override

## Public Attributes

- std::array< void const \*, sizeof...(MatcherTs)> **m\_matchers**

## Friends

- template<typename... MatchersRHS>  
[MatchAnyOfGeneric](#)< MatcherTs..., MatchersRHS... > **operator||** ([MatchAnyOfGeneric](#)< MatcherTs... > &&lhs, [MatchAnyOfGeneric](#)< MatchersRHS... > &&rhs)  
*Avoids type nesting for GenericAnyOf || GenericAnyOf case.*
- template<typename MatcherRHS >  
std::enable\_if\_t< is\_matcher< MatcherRHS >::value, [MatchAnyOfGeneric](#)< MatcherTs..., MatcherRHS > > **operator||** ([MatchAnyOfGeneric](#)< MatcherTs... > &&lhs, MatcherRHS const &rhs)  
*Avoids type nesting for GenericAnyOf || some matcher case.*
- template<typename MatcherLHS >  
std::enable\_if\_t< is\_matcher< MatcherLHS >::value, [MatchAnyOfGeneric](#)< MatcherLHS, MatcherTs... > > **operator||** (MatcherLHS const &lhs, [MatchAnyOfGeneric](#)< MatcherTs... > &&rhs)  
*Avoids type nesting for some matcher || GenericAnyOf case.*

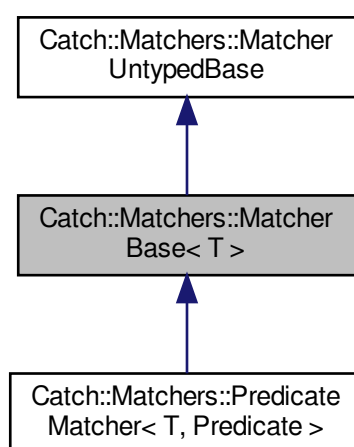
## Additional Inherited Members

The documentation for this class was generated from the following file:

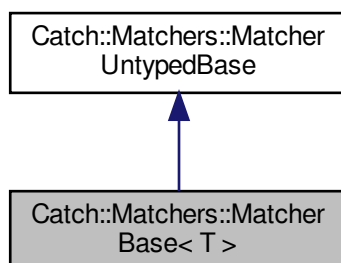
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.152 Catch::Matchers::MatcherBase< T > Class Template Reference

Inheritance diagram for Catch::Matchers::MatcherBase< T >:



Collaboration diagram for Catch::Matchers::MatcherBase< T >:



## Public Member Functions

- virtual bool **match** (T const &arg) const =0

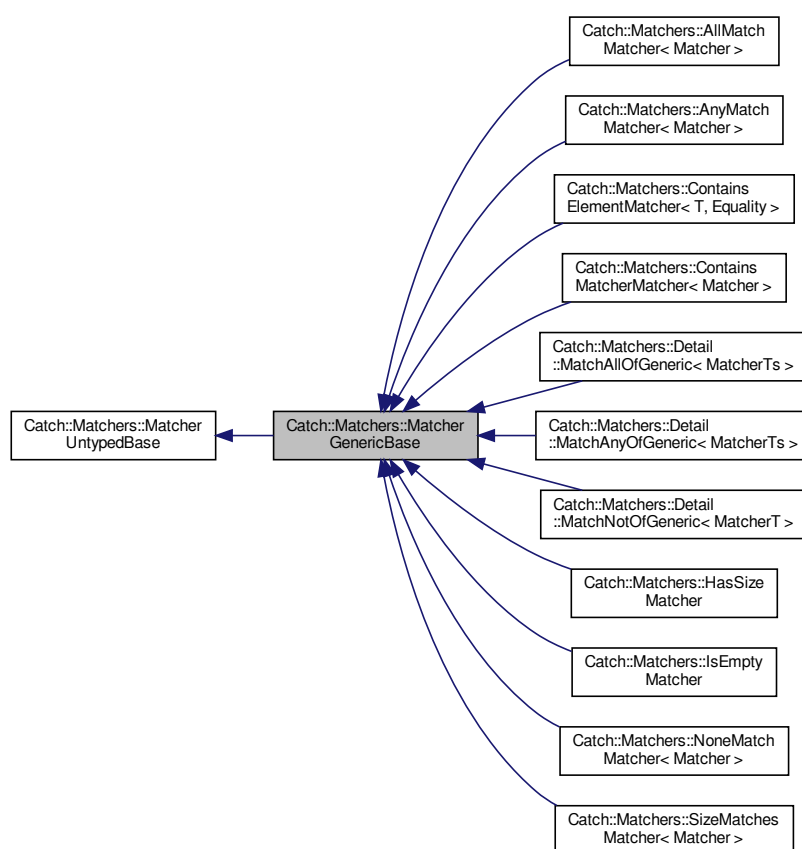
## Additional Inherited Members

The documentation for this class was generated from the following file:

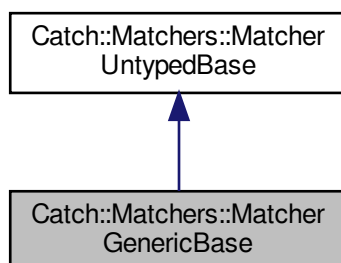
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.153 Catch::Matchers::MatcherGenericBase Class Reference

Inheritance diagram for Catch::Matchers::MatcherGenericBase:



Collaboration diagram for Catch::Matchers::MatcherGenericBase:



## Public Member Functions

- **MatcherGenericBase** ([MatcherGenericBase](#) &)=default
- **MatcherGenericBase** ([MatcherGenericBase](#) &&)=default
- [MatcherGenericBase](#) & **operator=** ([MatcherGenericBase](#) const &)=delete
- [MatcherGenericBase](#) & **operator=** ([MatcherGenericBase](#) &&)=delete

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



- [MatcherUntypedBase](#) & **operator=** ([MatcherUntypedBase](#) const &)=delete
- [MatcherUntypedBase](#) & **operator=** ([MatcherUntypedBase](#) &&)=delete
- std::string **toString** () const

### Protected Member Functions

- virtual std::string **describe** () const =0

### Protected Attributes

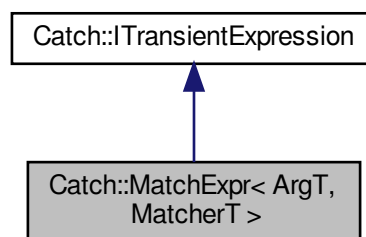
- std::string **m\_cachedToString**

The documentation for this class was generated from the following file:

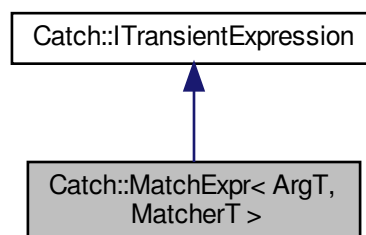
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.155 Catch::MatchExpr< ArgT, MatcherT > Class Template Reference

Inheritance diagram for Catch::MatchExpr< ArgT, MatcherT >:



Collaboration diagram for Catch::MatchExpr< ArgT, MatcherT >:



## Public Member Functions

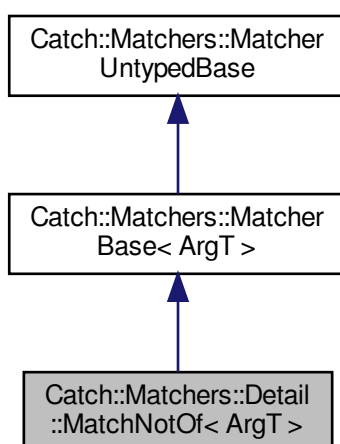
- **MatchExpr** (ArgT &&arg, MatcherT const &matcher, [StringRef](#) matcherString)
- void **streamReconstructedExpression** (std::ostream &os) const override

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

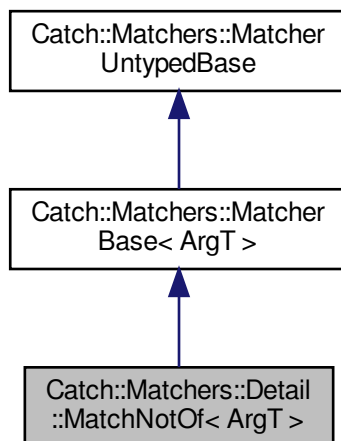
## 5.156 Catch::Matchers::Detail::MatchNotOf< ArgT > Class Template Reference

Inheritance diagram for Catch::Matchers::Detail::MatchNotOf< ArgT >:





Collaboration diagram for Catch::Matchers::Detail::MatchNotOf< ArgT >:



## Public Member Functions

- **MatchNotOf** ([MatcherBase](#)< ArgT > const &underlyingMatcher)
- bool **match** (ArgT const &arg) const override
- std::string **describe** () const override

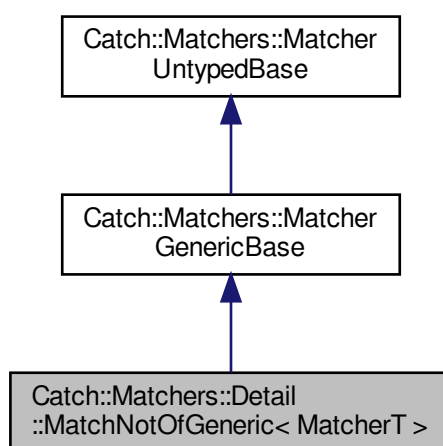
## Additional Inherited Members

The documentation for this class was generated from the following file:

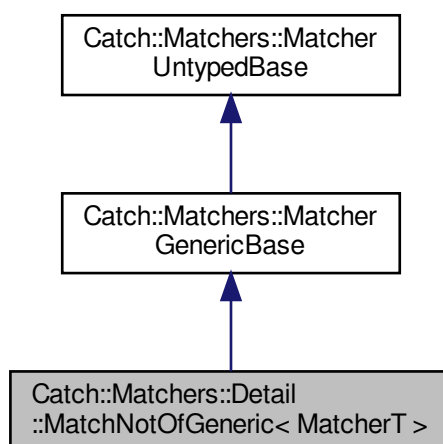
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.157 Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT > Class Template Reference

Inheritance diagram for Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT >:



Collaboration diagram for Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT >:



### Public Member Functions

- `MatchNotOfGeneric` (`MatchNotOfGeneric` const &)=delete

- [MatchNotOfGeneric](#) & **operator=** ([MatchNotOfGeneric](#) const &)=delete
- **MatchNotOfGeneric** ([MatchNotOfGeneric](#) &&)=default
- [MatchNotOfGeneric](#) & **operator=** ([MatchNotOfGeneric](#) &&)=default
- **MatchNotOfGeneric** (MatcherT const &matcher)
- template<typename Arg >  
bool **match** (Arg &&arg) const
- std::string **describe** () const override

## Friends

- MatcherT const & **operator!** ([MatchNotOfGeneric](#)< MatcherT > const &matcher)  
*Negating negation can just unwrap and return underlying matcher.*

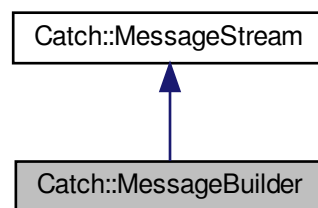
## Additional Inherited Members

The documentation for this class was generated from the following file:

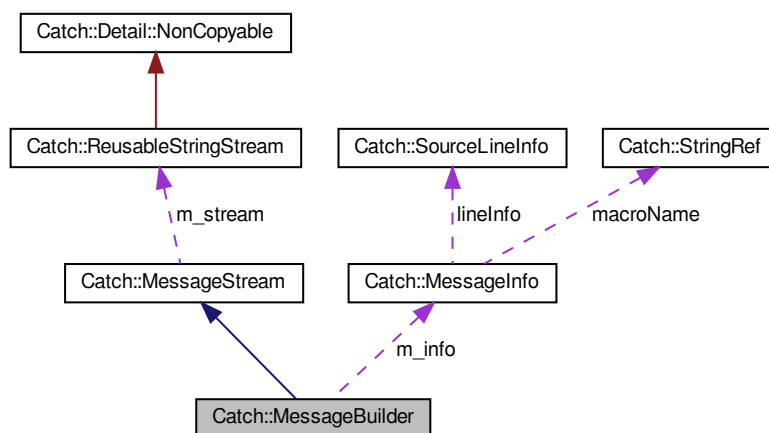
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.158 Catch::MessageBuilder Struct Reference

Inheritance diagram for Catch::MessageBuilder:



Collaboration diagram for `Catch::MessageBuilder`:



## Public Member Functions

- **MessageBuilder** ([StringRef](#) macroName, [SourceLineInfo](#) const &lineInfo, ResultWas::OfType type)
- `template<typename T >`  
[MessageBuilder](#) & **operator**<< (T const &value)

## Public Attributes

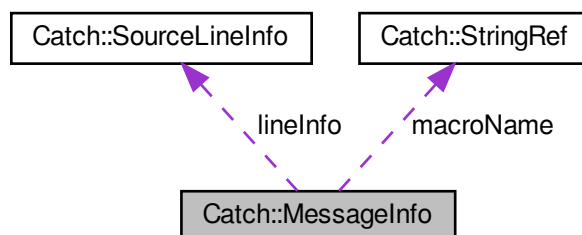
- [MessageInfo](#) `m_info`

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.159 Catch::MessageInfo Struct Reference

Collaboration diagram for `Catch::MessageInfo`:



## Public Member Functions

- **MessageInfo** ([StringRef](#) \_macroName, [SourceLineInfo](#) const &\_lineInfo, ResultWas::OfType \_type)
- bool **operator==** ([MessageInfo](#) const &other) const
- bool **operator<** ([MessageInfo](#) const &other) const

## Public Attributes

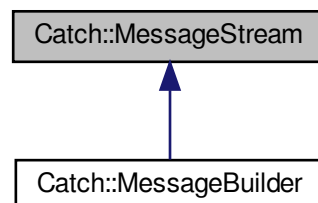
- [StringRef](#) **macroName**
- std::string **message**
- [SourceLineInfo](#) **lineInfo**
- ResultWas::OfType **type**
- unsigned int **sequence**

The documentation for this struct was generated from the following file:

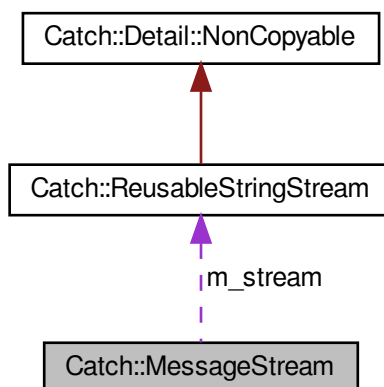
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.160 Catch::MessageStream Struct Reference

Inheritance diagram for Catch::MessageStream:



Collaboration diagram for `Catch::MessageStream`:



## Public Member Functions

- `template<typename T >`  
`MessageStream & operator<< (T const &value)`

## Public Attributes

- `ReusableStringStream m_stream`

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.161 Model< TSeq > Class Template Reference

Core class of epiworld.

```
#include <model-bones.hpp>
```

## Public Member Functions

- [DataBase](#)< TSeq > & **get\_db** ()
- `epiworld_double` & **operator()** (std::string pname)
- `size_t` **size** () const
- `size_t` **get\_n\_variants** () const
- `size_t` **get\_n\_tools** () const
- unsigned int **get\_ndays** () const
- unsigned int **get\_n\_replicates** () const
- void **set\_ndays** (unsigned int ndays)
- bool **get\_verbose** () const
- void **verbose\_off** ()
- void **verbose\_on** ()
- int [today](#) () const  
*The current time of the model.*
- void [write\\_data](#) (std::string fn\_variant\_info, std::string fn\_variant\_hist, std::string fn\_tool\_info, std::string fn↵\_tool\_hist, std::string fn\_total\_hist, std::string fn\_transmission, std::string fn\_transition) const  
*Wrapper of DataBase::write\_data*
- std::map< std::string, `epiworld_double` > & **params** ()
- void [reset](#) ()  
*Reset the model.*
- void **print** () const
- [Model](#)< TSeq > && **clone** () const
- void **get\_elapsed** (std::string unit="auto", `epiworld_double` \*last\_elapsed=nullptr, `epiworld_double` \*total\_↵elapsed=nullptr, std::string \*unit\_abbr=nullptr, bool print=true) const
- void [add\\_global\\_action](#) (std::function< void([Model](#)< TSeq > \*)> fun, int date=-99)  
*Set a global action.*
- void **run\_global\_actions** ()
- void **clear\_status\_set** ()
- const std::vector< `VirusPtr`< TSeq > > & **get\_viruses** () const
- const std::vector< `ToolPtr`< TSeq > > & **get\_tools** () const

### Set the backup object

*backup* can be used to restore the entire object after a run. This can be useful if the user wishes to have individuals start with the same network from the beginning.

- void **set\_backup** ()
- void **restore\_backup** ()

### Random number generation

#### Parameters

eng	<i>Random number generator</i>
s	<i>Seed</i>

- void **set\_rand\_engine** (std::mt19937 &eng)
- std::mt19937 \* **get\_rand\_engine** ()
- void **seed** (unsigned int s)
- void **set\_rand\_gamma** (`epiworld_double` alpha, `epiworld_double` beta)
- `epiworld_double` **runif** ()
- `epiworld_double` **rnorm** ()
- `epiworld_double` **rnorm** (`epiworld_double` mean, `epiworld_double` sd)

- `epiworld_double rgamma ()`
- `epiworld_double rgamma (epiworld_double alpha, epiworld_double beta)`

### Add Virus/Tool to the model

*This is done before the model has been initialized.*

#### Parameters

v	<i>Virus to be added</i>
t	<i>Tool to be added</i>
preval	<i>Initial prevalence (initial state.) It can be specified as a proportion (between zero and one,) or an integer indicating number of individuals.</i>

- `void add_virus (Virus< TSeq > v, epiworld_double preval)`
- `void add_virus_n (Virus< TSeq > v, unsigned int preval)`
- `void add_tool (Tool< TSeq > t, epiworld_double preval)`
- `void add_tool_n (Tool< TSeq > t, unsigned int preval)`

### Accessing population of the model

#### Parameters

fn	<i>std::string Filename of the edgelist file.</i>
skip	<i>int Number of lines to skip in fn.</i>
directed	<i>bool Whether the graph is directed or not.</i>
size	<i>Size of the network.</i>
al	<i>AdjList to read into the model.</i>

- `void agents_from_adjlist (std::string fn, int size, int skip=0, bool directed=false)`
- `void agents_from_adjlist (AdjList al)`
- `bool is_directed () const`
- `std::vector< Agent< TSeq > > * get_agents ()`
- `void agents_smallworld (unsigned int n=1000, unsigned int k=5, bool d=false, epiworld_double p=.01)`

### Functions to run the model

#### Parameters

seed	<i>Seed to be used for Pseudo-RNG.</i>
ndays	<i>Number of days (steps) of the simulation.</i>
fun	<i>In the case of run_multiple, a function that is called after each experiment.</i>

- `void init (unsigned int ndays, unsigned int seed)`
- `void update_status ()`
- `void mutate_variant ()`
- `void next ()`
- `void run ()`  
*Runs the simulation (after initialization)*
- `void run_multiple (unsigned int nexperiments, std::function< void(size_t, Model< TSeq > *)> fun=save←_run< TSeq >(), bool reset=true, bool verbose=true)`

### Rewire the network preserving the degree sequence.



This implementation assumes an undirected network, thus if  $\{(i,j), (k,l)\} \rightarrow \{(i,l), (k,j)\}$ , the reciprocal is also true, i.e.,  $\{(j,i), (l,k)\} \rightarrow \{(j,k), (l,i)\}$ .

#### Parameters

proportion	Proportion of ties to be rewired.
------------	-----------------------------------

#### Returns

A rewired version of the network.

- void **set\_rewire\_fun** (std::function< void(std::vector< Agent< TSeq >> \*, Model< TSeq > \*,  
epiworld\_double)> fun)
- void **set\_rewire\_prop** (epiworld\_double prop)
- epiworld\_double **get\_rewire\_prop** () const
- void **rewire** ()

### Export the network data in edgelist form

#### Parameters

fn	std::string. File name.
source	Integer vector
target	Integer vector

When passing the source and target, the function will write the edgelist on those.

- void **write\_edgelist** (std::string fn) const
- void **write\_edgelist** (std::vector< unsigned int > &source, std::vector< unsigned int > &target) const

### Manage status (states) in the model

The functions `get_status` return the current values for the statuses included in the model.

#### Parameters

lab	std::string Name of the status.
-----	---------------------------------

#### Returns

`add_status*` returns nothing.  
`get_status_*` returns a vector of pairs with the statuses and their labels.

- void **add\_status** (std::string lab, UpdateFun< TSeq > fun=nullptr)
- const std::vector< std::string > & **get\_status** () const
- const std::vector< UpdateFun< TSeq > > & **get\_status\_fun** () const
- void **print\_status\_codes** () const

### Set the user data object

#### Parameters

names	string vector with the names of the variables.
-------	--

- void **set\_user\_data** (std::vector< std::string > names)

- void **add\_user\_data** (unsigned int j, epiworld\_double x)
- void **add\_user\_data** (std::vector< epiworld\_double > x)
- [UserData](#)< TSeq > & **get\_user\_data** ()

### Queuing system

When queueing is on, the model will keep track of which agents are either in risk of exposure or exposed. This then is used at each step to act only on the aforementioned agents.

- void [queueing\\_on](#) ()  
Activates the queueing system (default.)
- void [queueing\\_off](#) ()  
Deactivates the queueing system.
- bool [is\\_queueing\\_on](#) () const  
Query if the queueing system is on.
- [Queue](#)< TSeq > & [get\\_queue](#) ()  
Retrieve the [Queue](#) object.

### Get the susceptibility reduction object

#### Parameters

v	
---	--

#### Returns

*epiworld\_double*

- void **set\_susceptibility\_reduction\_mixer** (MixerFun< TSeq > fun)
- void **set\_transmission\_reduction\_mixer** (MixerFun< TSeq > fun)
- void **set\_recovery\_enhancer\_mixer** (MixerFun< TSeq > fun)
- void **set\_death\_reduction\_mixer** (MixerFun< TSeq > fun)

### Friends

- class **Agent**< TSeq >
- class **DataBase**< TSeq >
- class **Queue**< TSeq >

### Tool Mixers

These functions combine the effects tools have to deliver a single effect. For example, wearing a mask, been vaccinated, and the immune system combine together to jointly reduce the susceptibility for a given virus.

- std::vector< epiworld\_double > **array\_double\_tmp**
- std::vector< [Virus](#)< TSeq > \* > **array\_virus\_tmp**
- **Model** ()
- **Model** (const [Model](#)< TSeq > &m)
- **Model** ([Model](#)< TSeq > &&m)
- [Model](#)< TSeq > & **operator=** (const [Model](#)< TSeq > &m)
- void **clone\_population** (std::vector< [Agent](#)< TSeq > > &p, bool &d, [Model](#)< TSeq > \*m=nullptr) const
- void **clone\_population** (const [Model](#)< TSeq > &m)

## Setting and accessing parameters from the model

[Tools](#) can incorporate parameters included in the model. Internally, parameters in the tool are stored as pointers to an `std::map<>` of parameters in the model. Using the `unsigned int` method directly fetches the parameters in the order these were added to the tool. Accessing parameters via the `std::string` method involves searching the parameter directly in the `std::map<>` member of the model (so it is not recommended.)

The function `set_param()` can be used when the parameter already exists in the model.

The `par()` function members are aliases for `get_param()`.

### Parameters

<i>initial_val</i>	
<i>pname</i>	Name of the parameter to add or to fetch

### Returns

The current value of the parameter in the model.

- `epiworld_double * p0`
- `epiworld_double * p1`
- `epiworld_double * p2`
- `epiworld_double * p3`
- `epiworld_double * p4`
- `epiworld_double * p5`
- `epiworld_double * p6`
- `epiworld_double * p7`
- `epiworld_double * p8`
- `epiworld_double * p9`
- `epiworld_double * p10`
- `epiworld_double * p11`
- `epiworld_double * p12`
- `epiworld_double * p13`
- `epiworld_double * p14`
- `epiworld_double * p15`
- `epiworld_double * p16`
- `epiworld_double * p17`
- `epiworld_double * p18`
- `epiworld_double * p19`
- `epiworld_double * p20`
- `epiworld_double * p21`
- `epiworld_double * p22`
- `epiworld_double * p23`
- `epiworld_double * p24`
- `epiworld_double * p25`
- `epiworld_double * p26`
- `epiworld_double * p27`
- `epiworld_double * p28`
- `epiworld_double * p29`
- `epiworld_double * p30`
- `epiworld_double * p31`
- `epiworld_double * p32`

- `epiworld_double * p33`
- `epiworld_double * p34`
- `epiworld_double * p35`
- `epiworld_double * p36`
- `epiworld_double * p37`
- `epiworld_double * p38`
- `epiworld_double * p39`
- `unsigned int npar_used = 0u`
- `epiworld_double add_param` (`epiworld_double initial_val`, `std::string pname`)
- `epiworld_double set_param` (`std::string pname`)
- `epiworld_double get_param` (`unsigned int k`)
- `epiworld_double get_param` (`std::string pname`)
- `epiworld_double par` (`unsigned int k`)
- `epiworld_double par` (`std::string pname`)

### 5.161.1 Detailed Description

```
template<typename TSeq = int>
class Model< TSeq >
```

Core class of epiworld.

The model class provides the wrapper that puts together [Agent](#), [Virus](#), and [Tools](#).

#### Template Parameters

<i>TSeq</i>	Type of sequence. In principle, users can build models in which virus and human sequence is represented as numeric vectors (if needed.)
-------------	---

### 5.161.2 Member Function Documentation

#### 5.161.2.1 add\_global\_action()

```
template<typename TSeq = int>
void Model< TSeq >::add_global_action (
    std::function< void(Model< TSeq > *)> fun,
    int date = -99 )
```

Set a global action.

#### Parameters

<i>fun</i>	A function to be called on the prescribed dates
<i>date</i>	Integer indicating when the function is called (see details)

When date is less than zero, then the function is called at the end of every day. Otherwise, the function will be called

only at the end of the indicated date.

### 5.161.2.2 reset()

```
template<typename TSeq = int>
void Model< TSeq >::reset ( )
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

### 5.161.2.3 run\_multiple()

```
template<typename TSeq = int>
void Model< TSeq >::run_multiple (
    unsigned int nexperiments,
    std::function< void(size_t, Model< TSeq > *)> fun = save_run< TSeq >(),
    bool reset = true,
    bool verbose = true )
```

#### Parameters

<i>nexperiments</i>	Multiple runs of the simulation
---------------------	---------------------------------

### 5.161.2.4 write\_data()

```
template<typename TSeq = int>
void Model< TSeq >::write_data (
    std::string fn_variant_info,
    std::string fn_variant_hist,
    std::string fn_tool_info,
    std::string fn_tool_hist,
    std::string fn_total_hist,
    std::string fn_transmission,
    std::string fn_transition ) const
```

Wrapper of `DataBase::write_data`

## Parameters

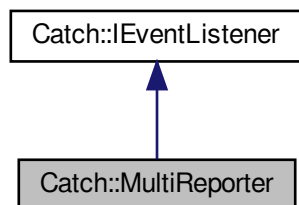
<i>fn_variant_info</i>	Filename. Information about the variant.
<i>fn_variant_hist</i>	Filename. History of the variant.
<i>fn_tool_info</i>	Filename. Information about the tool.
<i>fn_tool_hist</i>	Filename. History of the tool.
<i>fn_total_hist</i>	Filename. Aggregated history (status)
<i>fn_transmission</i>	Filename. Transmission history.
<i>fn_transition</i>	Filename. Markov transition history.

The documentation for this class was generated from the following files:

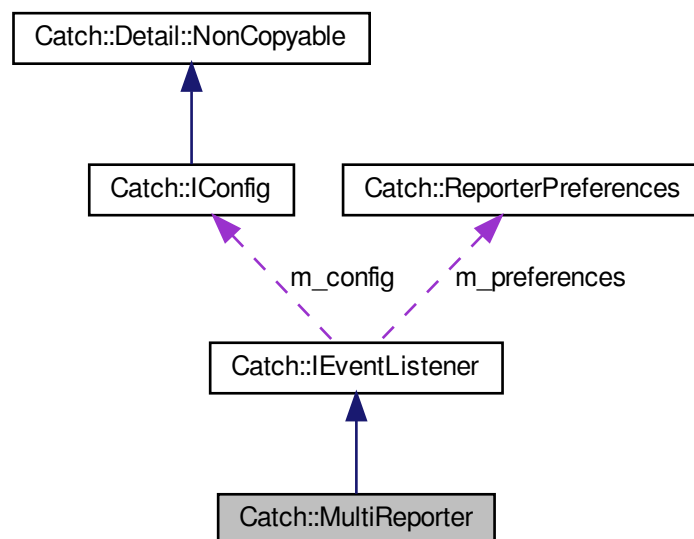
- include/epiworld/agent-meat-status.hpp
- include/epiworld/model-bones.hpp

## 5.162 Catch::MultiReporter Class Reference

Inheritance diagram for Catch::MultiReporter:



Collaboration diagram for Catch::MultiReporter:



## Public Member Functions

- void **addListener** (IEventListenerPtr &&listener)
- void **addReporter** (IEventListenerPtr &&reporter)
- void **noMatchingTestCases** (StringRef unmatchedSpec) override  
*Called when no test cases match provided test spec.*
- void **fatalErrorEncountered** (StringRef error) override  
*Called if a fatal error (signal/structured exception) occurred.*
- void **reportInvalidTestSpec** (StringRef arg) override  
*Called for all invalid test specs from the cli.*
- void **benchmarkPreparing** (StringRef name) override  
*Called when user-code is being probed before the actual benchmark runs.*
- void **benchmarkStarting** (BenchmarkInfo const &benchmarkInfo) override  
*Called after probe but before the user-code is being benchmarked.*
- void **benchmarkEnded** (BenchmarkStats<> const &benchmarkStats) override  
*Called with the benchmark results if benchmark successfully finishes.*
- void **benchmarkFailed** (StringRef error) override  
*Called if running the benchmarks fails for any reason.*
- void **testRunStarting** (TestRunInfo const &testRunInfo) override
- void **testCaseStarting** (TestCaseInfo const &testInfo) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void **testCasePartialStarting** (TestCaseInfo const &testInfo, uint64\_t partNumber) override  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- void **sectionStarting** (SectionInfo const &sectionInfo) override  
*Called when a SECTION is being entered. Not called for skipped sections.*
- void **assertionStarting** (AssertionInfo const &assertionInfo) override

- *Called before assertion success/failure is evaluated.*
- void [assertionEnded](#) ([AssertionStats](#) const &assertionStats) override  
*Called after assertion was fully evaluated.*
- void [sectionEnded](#) ([SectionStats](#) const &sectionStats) override  
*Called after a `SECTION` has finished running.*
- void [testCasePartialEnded](#) ([TestCaseStats](#) const &testInfo, uint64\_t partNumber) override  
*Called every time a `TEST_CASE` is entered, including repeats (due to sections)*
- void [testCaseEnded](#) ([TestCaseStats](#) const &testCaseStats) override  
*Called once for each `TEST_CASE`, no matter how many times it is entered.*
- void [testRunEnded](#) ([TestRunStats](#) const &testRunStats) override
- void [skipTest](#) ([TestCaseInfo](#) const &testInfo) override  
*Called with test cases that are skipped due to the test run aborting.*
- void [listReporters](#) (std::vector< [ReporterDescription](#) > const &descriptions) override  
*Writes out information about provided reporters using reporter-specific format.*
- void [listListeners](#) (std::vector< [ListenerDescription](#) > const &descriptions) override  
*Writes out the provided listeners descriptions using reporter-specific format.*
- void [listTests](#) (std::vector< [TestCaseHandle](#) > const &tests) override  
*Writes out information about provided tests using reporter-specific format.*
- void [listTags](#) (std::vector< [TagInfo](#) > const &tags) override  
*Writes out information about the provided tags using reporter-specific format.*
- [IEventListener](#) ([IConfig](#) const \*config)

## Additional Inherited Members

### 5.162.1 Member Function Documentation

#### 5.162.1.1 testRunEnded()

```
void Catch::MultiReporter::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Implements [Catch::IEventListener](#).

#### 5.162.1.2 testRunStarting()

```
void Catch::MultiReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Implements [Catch::IEventListener](#).

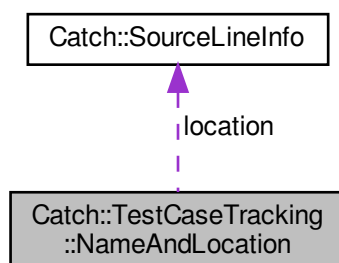
The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)



## 5.163 Catch::TestCaseTracking::NameAndLocation Struct Reference

Collaboration diagram for Catch::TestCaseTracking::NameAndLocation:



### Public Member Functions

- **NameAndLocation** (std::string const &\_name, [SourceLineInfo](#) const &\_location)

### Public Attributes

- std::string **name**
- [SourceLineInfo](#) **location**

### Friends

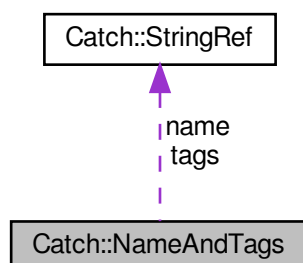
- bool **operator==** ([NameAndLocation](#) const &lhs, [NameAndLocation](#) const &rhs)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.164 Catch::NameAndTags Struct Reference

Collaboration diagram for Catch::NameAndTags:



### Public Member Functions

- constexpr **NameAndTags** ([StringRef](#) name\_<sub>=</sub>[StringRef](#)(), [StringRef](#) tags\_<sub>=</sub>[StringRef](#)()) noexcept

### Public Attributes

- [StringRef](#) name
- [StringRef](#) tags

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.165 Catch::CumulativeReporterBase::Node< T, ChildNodeT > Struct Template Reference

### Public Types

- using **ChildNodes** = std::vector< [Detail::unique\\_ptr](#)< ChildNodeT > >

### Public Member Functions

- **Node** (T const &\_value)

## Public Attributes

- T **value**
- ChildNodes **children**

The documentation for this struct was generated from the following file:

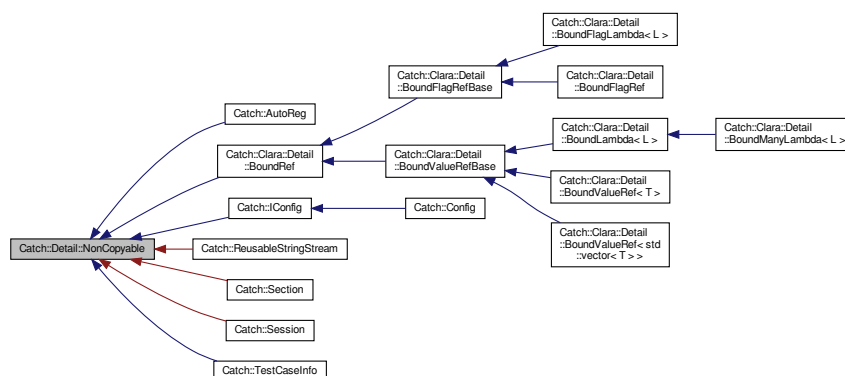
- include/catch2/catch\_amalgamated.hpp

## 5.166 Catch::Detail::NonCopyable Class Reference

Deriving classes become noncopyable and nonmovable.

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for Catch::Detail::NonCopyable:



### 5.166.1 Detailed Description

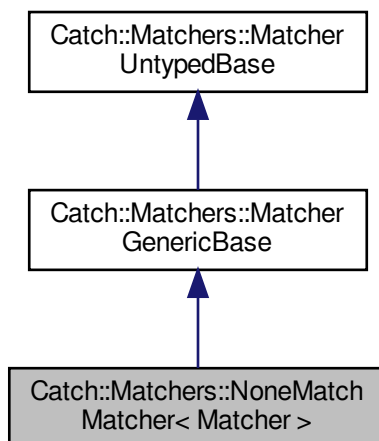
Deriving classes become noncopyable and nonmovable.

The documentation for this class was generated from the following file:

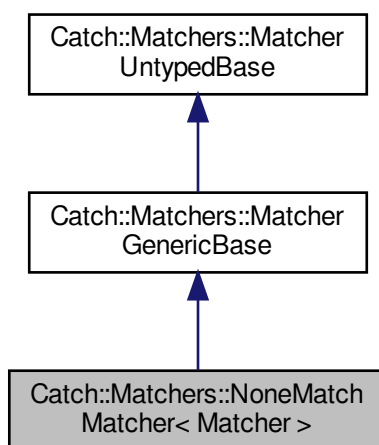
- include/catch2/catch\_amalgamated.hpp

## 5.167 Catch::Matchers::NoneMatcher< Matcher > Class Template Reference

Inheritance diagram for Catch::Matchers::NoneMatcher< Matcher >:



Collaboration diagram for Catch::Matchers::NoneMatcher< Matcher >:



### Public Member Functions

- **NoneMatcher** (Matcher matcher)
- std::string **describe** () const override
- template<typename RangeLike >  
bool **match** (RangeLike &&rng) const

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.168 Catch::Benchmark::now< Clock > Struct Template Reference

### Public Member Functions

- TimePoint< Clock > **operator()** () const

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.169 Catch::Benchmark::Detail::ObjectStorage< T, Destruct > Struct Template Reference

### Public Member Functions

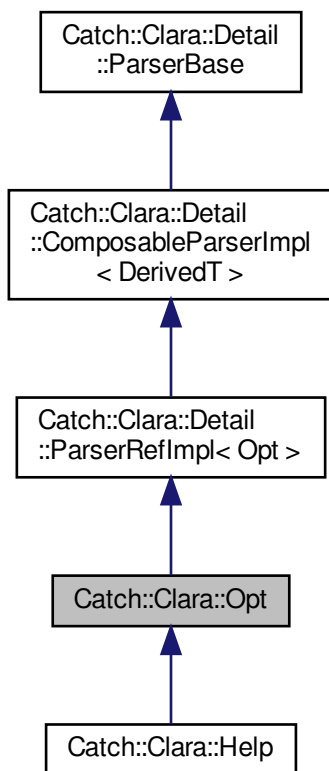
- **ObjectStorage** (const [ObjectStorage](#) &other)
- **ObjectStorage** ([ObjectStorage](#) &&other)
- template<typename... Args>  
void **construct** (Args &&... args)
- template<bool AllowManualDestruction = !Destruct>  
std::enable\_if\_t< AllowManualDestruction > **destruct** ()

The documentation for this struct was generated from the following file:

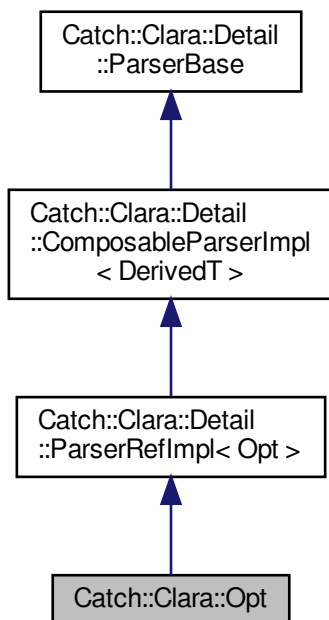
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.170 Catch::Clara::Opt Class Reference

Inheritance diagram for Catch::Clara::Opt:



Collaboration diagram for Catch::Clara::Opt:



## Public Member Functions

- `template<typename LambdaT >`  
`Opt` (LambdaT const &ref)
- `Opt` (bool &ref)
- `template<typename LambdaT , typename = typename std::enable_if_t< Detail::is_unary_function<LambdaT>::value>>`  
`Opt` (LambdaT const &ref, std::string const &hint)
- `template<typename LambdaT >`  
`Opt` ([accept\\_many\\_t](#), LambdaT const &ref, std::string const &hint)
- `template<typename T , typename = typename std::enable_if_t< !Detail::is_unary_function<T>::value>>`  
`Opt` (T &ref, std::string const &hint)
- `auto operator[]` (std::string const &optName) -> `Opt` &
- `std::vector< Detail::HelpColumns > getHelpColumns` () const
- `bool isMatch` (std::string const &optToken) const
- `Detail::InternalParseResult parse` (std::string const &, [Detail::TokenStream](#) const &tokens) const override
- `Detail::Result validate` () const override

## Protected Attributes

- `std::vector< std::string > m_optNames`

## Additional Inherited Members

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.171 Catch::Optional< T > Class Template Reference

### Public Member Functions

- **Optional** (T const &\_value)
- **Optional** ([Optional](#) const &\_other)
- [Optional](#) & **operator=** ([Optional](#) const &\_other)
- [Optional](#) & **operator=** (T const &\_value)
- void **reset** ()
- T & **operator\*** ()
- T const & **operator\*** () const
- T \* **operator->** ()
- const T \* **operator->** () const
- T **valueOr** (T const &defaultValue) const
- bool **some** () const
- bool **none** () const
- bool **operator!** () const
- **operator bool** () const

### Friends

- bool **operator==** ([Optional](#) const &a, [Optional](#) const &b)
- bool **operator!=** ([Optional](#) const &a, [Optional](#) const &b)

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.172 Catch::Benchmark::OutlierClassification Struct Reference

### Public Member Functions

- int **total** () const

### Public Attributes

- int **samples\_seen** = 0
- int **low\_severe** = 0
- int **low\_mild** = 0
- int **high\_mild** = 0
- int **high\_severe** = 0

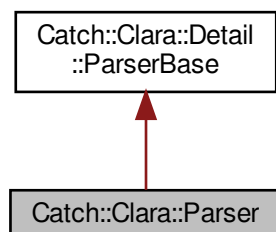
The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

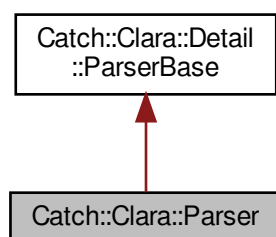


## 5.173 Catch::Clara::Parser Class Reference

Inheritance diagram for Catch::Clara::Parser:



Collaboration diagram for Catch::Clara::Parser:



### Public Member Functions

- auto **operator**|= (ExeName const &exeName) -> Parser &
- auto **operator**|= (Arg const &arg) -> Parser &
- auto **operator**|= (Opt const &opt) -> Parser &
- Parser & **operator**|= (Parser const &other)
- template<typename T >  
auto **operator**| (T const &other) const -> Parser
- std::vector< Detail::HelpColumns > **getHelpColumns** () const
- void **writeToStream** (std::ostream &os) const
- Detail::Result **validate** () const override
- Detail::InternalParseResult **parse** (std::string const &exeName, Detail::TokenStream const &tokens) const override

## Friends

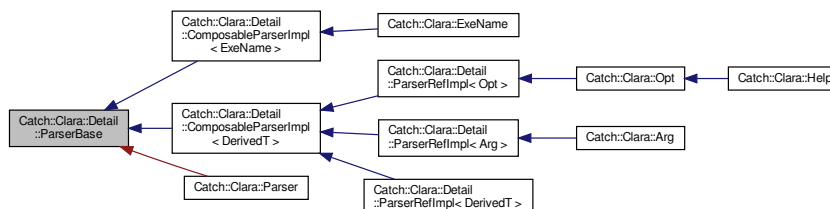
- auto **operator**<< (std::ostream &os, [Parser](#) const &parser) -> std::ostream &

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.174 Catch::Clara::Detail::ParserBase Class Reference

Inheritance diagram for Catch::Clara::Detail::ParserBase:



## Public Member Functions

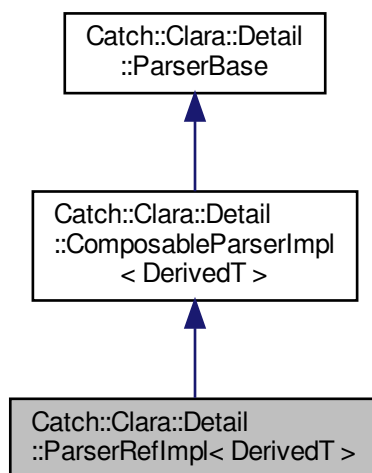
- virtual auto **validate** () const -> [Result](#)
- virtual auto **parse** (std::string const &exeName, [TokenStream](#) const &tokens) const -> [InternalParseResult](#)=0
- virtual size\_t **cardinality** () const
- [InternalParseResult](#) **parse** ([Args](#) const &args) const

The documentation for this class was generated from the following file:

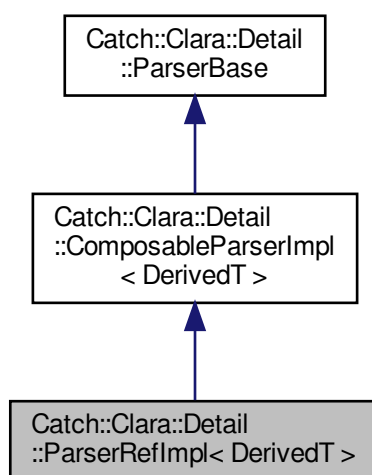
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.175 Catch::Clara::Detail::ParserRefImpl< DerivedT > Class Template Reference

Inheritance diagram for Catch::Clara::Detail::ParserRefImpl< DerivedT >:



Collaboration diagram for Catch::Clara::Detail::ParserRefImpl< DerivedT >:



### Public Member Functions

- `template<typename LambdaT >`  
**ParserRefImpl** ([accept\\_many\\_t](#), LambdaT const &ref, std::string const &hint)

- `template<typename T , typename = typename std::enable_if_t< !Detail::is_unary_function<T>::value>>`  
**ParserRefImpl** (T &ref, std::string const &hint)
- `template<typename LambdaT , typename = typename std::enable_if_t< Detail::is_unary_function<LambdaT>::value>>`  
**ParserRefImpl** (LambdaT const &ref, std::string const &hint)
- `auto operator()` (std::string const &description) -> DerivedT &
- `auto optional ()` -> DerivedT &
- `auto required ()` -> DerivedT &
- `auto isOptional ()` const -> bool
- `auto cardinality ()` const -> size\_t override
- `std::string const & hint ()` const

## Protected Member Functions

- **ParserRefImpl** (std::shared\_ptr< [BoundRef](#) > const &ref)

## Protected Attributes

- Optionality **m\_optionality** = Optionality::Optional
- std::shared\_ptr< [BoundRef](#) > **m\_ref**
- std::string **m\_hint**
- std::string **m\_description**

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.176 Catch::Clara::Detail::ParseState Class Reference

### Public Member Functions

- **ParseState** (ParseResultType type, [TokenStream](#) const &remainingTokens)
- ParseResultType **type** () const
- [TokenStream](#) const & **remainingTokens** () const

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.177 PersonTools< TSeq > Class Template Reference

The documentation for this class was generated from the following file:

- include/epiworld/config.hpp

## 5.178 Catch::pluralise Class Reference

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- constexpr **pluralise** (std::uint64\_t count, [StringRef](#) label)

### Friends

- std::ostream & **operator**<< (std::ostream &os, [pluralise](#) const &pluraliser)

### 5.178.1 Detailed Description

Helper for streaming a "count [maybe-plural-of-label]" human-friendly string

Usage example:

```
std::cout << "Found " << pluralise(count, "error") << '\n';
```

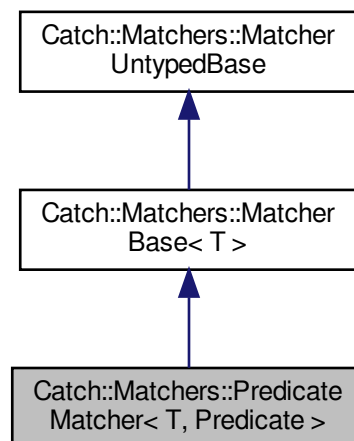
**Important:** The provided string must outlive the instance

The documentation for this class was generated from the following file:

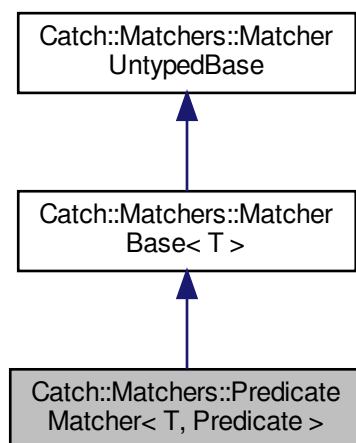
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.179 Catch::Matchers::PredicateMatcher< T, Predicate > Class Template Reference

Inheritance diagram for Catch::Matchers::PredicateMatcher< T, Predicate >:



Collaboration diagram for `Catch::Matchers::PredicateMatcher< T, Predicate >`:



## Public Member Functions

- **PredicateMatcher** (Predicate &&elem, std::string const &descr)
- bool **match** (T const &item) const override
- std::string **describe** () const override

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.180 Catch::ProcessedReporterSpec Struct Reference

```
#include <catch_amalgamated.hpp>
```

## Public Attributes

- std::string **name**
- std::string **outputFilename**
- [ColourMode](#) **colourMode**
- std::map< std::string, std::string > **customOptions**

## Friends

- bool **operator==** ([ProcessedReporterSpec](#) const &lhs, [ProcessedReporterSpec](#) const &rhs)
- bool **operator!=** ([ProcessedReporterSpec](#) const &lhs, [ProcessedReporterSpec](#) const &rhs)

### 5.180.1 Detailed Description

[ReporterSpec](#) but with the defaults filled in.

Like [ReporterSpec](#), the semantics are unchecked.

The documentation for this struct was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.181 Progress Class Reference

A simple progress bar.

```
#include <progress.hpp>
```

### Public Member Functions

- **Progress** (int n\_, int width\_)
- void **start** ()
- void **next** ()
- void **end** ()

### 5.181.1 Detailed Description

A simple progress bar.

The documentation for this class was generated from the following file:

- include/epiworld/progress.hpp

## 5.182 Queue< TSeq > Class Template Reference

Controls which agents are verified at each step.

```
#include <queue-bones.hpp>
```

### Public Member Functions

- void **operator+=** ([Agent](#)< TSeq > \*p)
- void **operator-=** ([Agent](#)< TSeq > \*p)
- epiworld\_fast\_int **operator[]** (unsigned int i) const
- void **set\_model** ([Model](#)< TSeq > \*m)

### 5.182.1 Detailed Description

```
template<typename TSeq = int>
class Queue< TSeq >
```

Controls which agents are verified at each step.

The idea is that only agents who are either in an infected state or have an infected neighbor should be checked. Otherwise it makes no sense (no chance to recover or capture the disease).

## Template Parameters

<i>TSeq</i>	
-------------	--

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/queue-bones.hpp

## 5.183 RandGraph Class Reference

### Public Member Functions

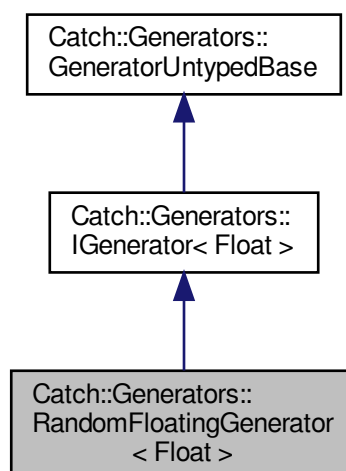
- **RandGraph** (int N\_)
- void **init** (int s)
- void **set\_rand\_engine** (std::mt19937 &e)
- epiworld\_double **runif** ()

The documentation for this class was generated from the following file:

- include/epiworld/random\_graph.hpp

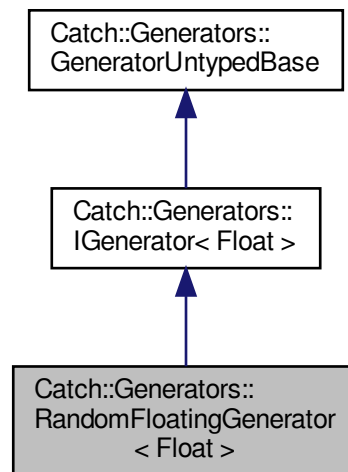
## 5.184 Catch::Generators::RandomFloatingGenerator< Float > Class Template Reference

Inheritance diagram for Catch::Generators::RandomFloatingGenerator< Float >:





Collaboration diagram for Catch::Generators::RandomFloatingGenerator< Float >:



## Public Member Functions

- **RandomFloatingGenerator** (Float a, Float b, std::uint32\_t seed)
- Float const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.184.1 Member Function Documentation

#### 5.184.1.1 next()

```
template<typename Float >
bool Catch::Generators::RandomFloatingGenerator< Float >::next ( ) [inline], [override],
[virtual]
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

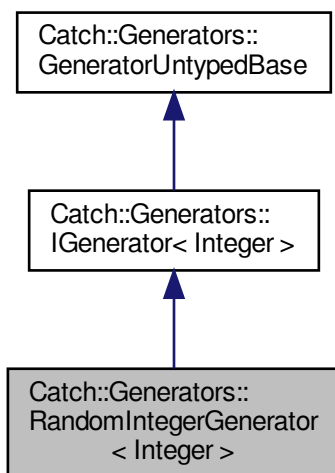
Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

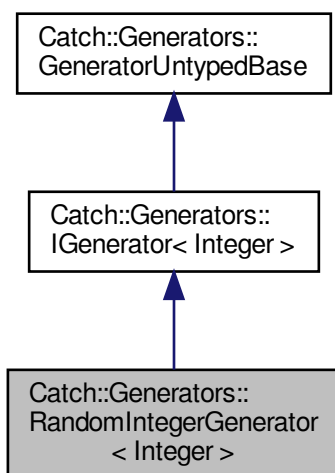
- include/catch2/catch\_amalgamated.hpp

## 5.185 Catch::Generators::RandomIntegerGenerator< Integer > Class Template Reference

Inheritance diagram for Catch::Generators::RandomIntegerGenerator< Integer >:



Collaboration diagram for Catch::Generators::RandomIntegerGenerator< Integer >:



### Public Member Functions

- **RandomIntegerGenerator** (Integer a, Integer b, std::uint32\_t seed)

- Integer const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.185.1 Member Function Documentation

#### 5.185.1.1 next()

```
template<typename Integer >  
bool Catch::Generators::RandomIntegerGenerator< Integer >::next ( ) [inline], [override],  
[virtual]
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

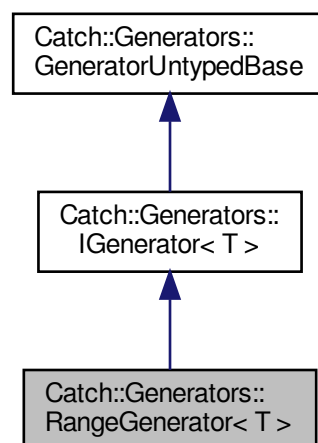
Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

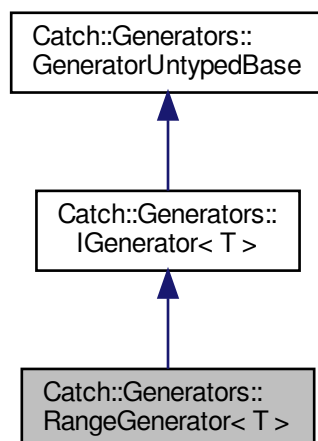
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.186 Catch::Generators::RangeGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::RangeGenerator< T >:



Collaboration diagram for `Catch::Generators::RangeGenerator< T >`:



## Public Member Functions

- **RangeGenerator** (T const &start, T const &end, T const &step)
- **RangeGenerator** (T const &start, T const &end)
- T const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.186.1 Member Function Documentation

#### 5.186.1.1 next()

```

template<typename T >
bool Catch::Generators::RangeGenerator< T >::next ( ) [inline], [override], [virtual]
  
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.187 Catch::ratio\_string< Ratio > Struct Template Reference

### Static Public Member Functions

- static std::string **symbol** ()

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.188 Catch::ratio\_string< std::atto > Struct Reference

### Static Public Member Functions

- static char **symbol** ()

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.189 Catch::ratio\_string< std::femto > Struct Reference

### Static Public Member Functions

- static char **symbol** ()

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.190 Catch::ratio\_string< std::micro > Struct Reference

### Static Public Member Functions

- static char **symbol** ()

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.191 Catch::ratio\_string< std::milli > Struct Reference

### Static Public Member Functions

- static char **symbol** ()

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.192 Catch::ratio\_string< std::nano > Struct Reference

### Static Public Member Functions

- static char **symbol** ()

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.193 Catch::ratio\_string< std::pico > Struct Reference

### Static Public Member Functions

- static char **symbol** ()

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.194 Catch::RedirectedStdErr Class Reference

### Public Member Functions

- auto **str** () const -> std::string

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.195 Catch::RedirectedStdOut Class Reference

### Public Member Functions

- auto **str** () const -> std::string

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.196 Catch::RedirectedStream Class Reference

### Public Member Functions

- **RedirectedStream** (std::ostream &originalStream, std::ostream &redirectionStream)

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.197 Catch::RedirectedStreams Class Reference

### Public Member Functions

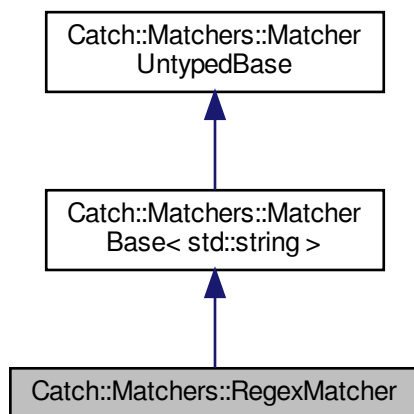
- **RedirectedStreams** ([RedirectedStreams](#) const &)=delete
- [RedirectedStreams](#) & **operator=** ([RedirectedStreams](#) const &)=delete
- **RedirectedStreams** ([RedirectedStreams](#) &&)=delete
- [RedirectedStreams](#) & **operator=** ([RedirectedStreams](#) &&)=delete
- **RedirectedStreams** (std::string &redirectedCout, std::string &redirectedCerr)

The documentation for this class was generated from the following file:

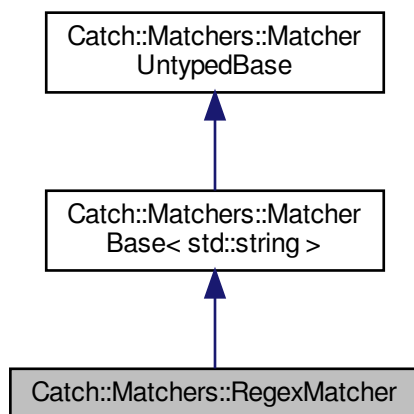
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.198 Catch::Matchers::RegexMatcher Class Reference

Inheritance diagram for Catch::Matchers::RegexMatcher:



Collaboration diagram for Catch::Matchers::RegexMatcher:



### Public Member Functions

- **RegexMatcher** (std::string regex, CaseSensitive caseSensitivity)
- bool **match** (std::string const &matchee) const override
- std::string **describe** () const override



## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.199 Catch::RegistrarForTagAliases Struct Reference

### Public Member Functions

- **RegistrarForTagAliases** (char const \*alias, char const \*tag, [SourceLineInfo](#) const &lineInfo)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.200 Catch::Benchmark::Detail::repeater< Fun > Struct Template Reference

### Public Member Functions

- void **operator()** (int k) const

### Public Attributes

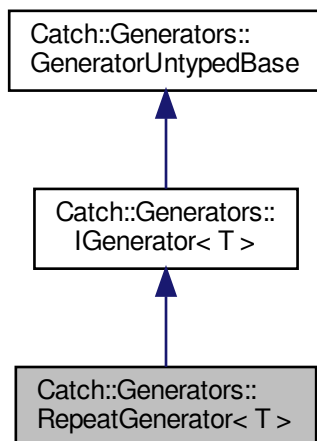
- Fun **fun**

The documentation for this struct was generated from the following file:

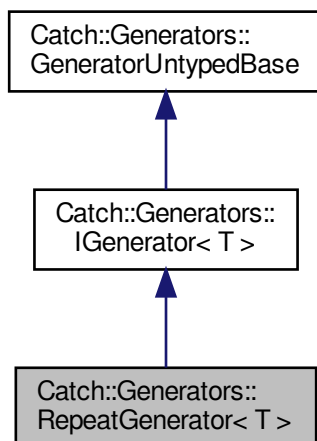
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.201 Catch::Generators::RepeatGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::RepeatGenerator< T >:



Collaboration diagram for Catch::Generators::RepeatGenerator< T >:



### Public Member Functions

- **RepeatGenerator** (size\_t repeats, [GeneratorWrapper< T >](#) &&generator)
- T const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.201.1 Member Function Documentation

#### 5.201.1.1 next()

```
template<typename T >
bool Catch::Generators::RepeatGenerator< T >::next ( ) [inline], [override], [virtual]
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

Implements [Catch::Generators::GeneratorUntypedBase](#).

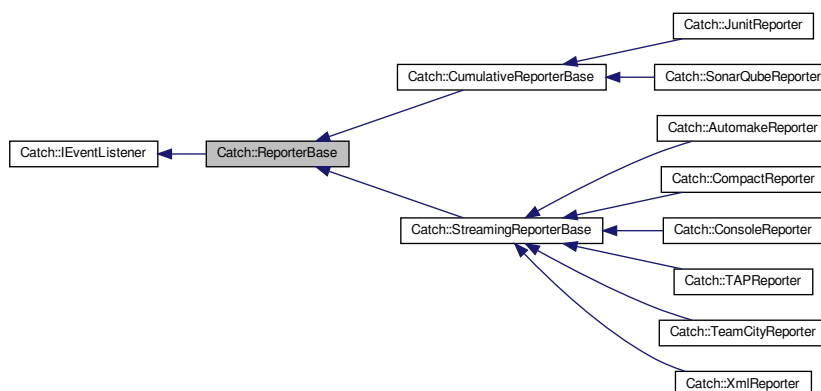
The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

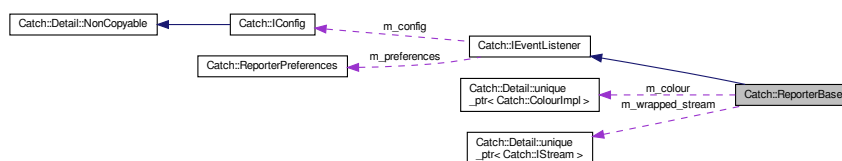
## 5.202 Catch::ReporterBase Class Reference

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for Catch::ReporterBase:



Collaboration diagram for Catch::ReporterBase:



## Public Member Functions

- **ReporterBase** ([ReporterConfig](#) &&config)
- void [listReporters](#) (std::vector< [ReporterDescription](#) > const &descriptions) override
- void [listListeners](#) (std::vector< [ListenerDescription](#) > const &descriptions) override
- void [listTests](#) (std::vector< [TestCaseHandle](#) > const &tests) override
- void [listTags](#) (std::vector< [TagInfo](#) > const &tags) override

## Protected Attributes

- [Detail::unique\\_ptr< IStream > m\\_wrapped\\_stream](#)  
*The stream wrapper as passed to us by outside code.*
- std::ostream & [m\\_stream](#)
- [Detail::unique\\_ptr< ColourImpl > m\\_colour](#)  
*Colour implementation this reporter was configured for.*
- std::map< std::string, std::string > [m\\_customOptions](#)  
*The custom reporter options user passed down to the reporter.*

### 5.202.1 Detailed Description

This is the base class for all reporters.

If are writing a reporter, you must derive from this type, or one of the helper reporter bases that are derived from this type.

[ReporterBase](#) centralizes handling of various common tasks in reporters, like storing the right stream for the reporters to write to, and providing the default implementation of the different listing events.

### 5.202.2 Member Function Documentation

#### 5.202.2.1 listListeners()

```
void Catch::ReporterBase::listListeners (
    std::vector< ListenerDescription > const & descriptions ) [override], [virtual]
```

Provides a simple default listing of listeners

Looks similarly to listing of reporters, but with listener type instead of reporter name.

Implements [Catch::IEventListener](#).

Reimplemented in [Catch::XmlReporter](#).

### 5.202.2.2 listReporters()

```
void Catch::ReporterBase::listReporters (
    std::vector< ReporterDescription > const & descriptions ) [override], [virtual]
```

Provides a simple default listing of reporters.

Should look roughly like the reporter listing in v2 and earlier versions of Catch2.

Implements [Catch::IEventListener](#).

Reimplemented in [Catch::XmlReporter](#).

### 5.202.2.3 listTags()

```
void Catch::ReporterBase::listTags (
    std::vector< TagInfo > const & tags ) [override], [virtual]
```

Provides a simple default listing of tags.

Should look roughly like the tag listing in v2 and earlier versions of Catch2.

Implements [Catch::IEventListener](#).

Reimplemented in [Catch::XmlReporter](#).

### 5.202.2.4 listTests()

```
void Catch::ReporterBase::listTests (
    std::vector< TestCaseHandle > const & tests ) [override], [virtual]
```

Provides a simple default listing of tests.

Should look roughly like the test listing in v2 and earlier versions of Catch2. Especially supports low-verbosity listing that mimics the old `--list-test-names-only` output.

Implements [Catch::IEventListener](#).

Reimplemented in [Catch::XmlReporter](#).

## 5.202.3 Member Data Documentation

### 5.202.3.1 m\_stream

```
std::ostream& Catch::ReporterBase::m_stream [protected]
```

Cached output stream from `m_wrapped_stream` to reduce number of indirect calls needed to write output.

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.203 Catch::ReporterConfig Struct Reference

### Public Member Functions

- **ReporterConfig** ([IConfig](#) const \* `_fullConfig`, [Detail::unique\\_ptr< IStream >](#) `_stream`, [ColourMode](#) `colourMode`, [std::map< std::string, std::string >](#) `customOptions`)
- **ReporterConfig** ([ReporterConfig](#) &&)=default
- [ReporterConfig](#) & **operator=** ([ReporterConfig](#) &&)=default
- [Detail::unique\\_ptr< IStream >](#) **takeStream** () &&
- [IConfig](#) const \* **fullConfig** () const
- [ColourMode](#) **colourMode** () const
- [std::map< std::string, std::string >](#) const & **customOptions** () const

The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.204 Catch::ReporterDescription Struct Reference

### Public Attributes

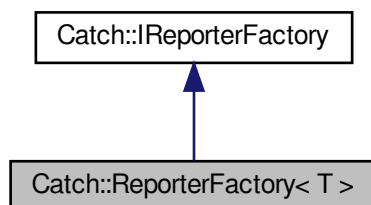
- [std::string](#) **name**
- [std::string](#) **description**

The documentation for this struct was generated from the following file:

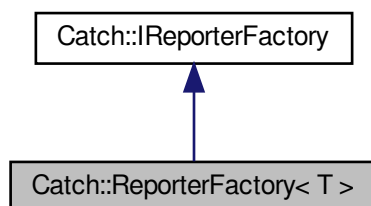
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.205 Catch::ReporterFactory< T > Class Template Reference

Inheritance diagram for Catch::ReporterFactory< T >:



Collaboration diagram for Catch::ReporterFactory< T >:



### Additional Inherited Members

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.206 Catch::ReporterPreferences Struct Reference

```
#include <catch_amalgamated.hpp>
```

### Public Attributes

- bool [shouldRedirectStdOut](#) = false
- bool [shouldReportAllAssertions](#) = false

### 5.206.1 Detailed Description

By setting up its preferences, a reporter can modify Catch2's behaviour in some regards, e.g. it can request Catch2 to capture writes to stdout/stderr during test execution, and pass them to the reporter.

### 5.206.2 Member Data Documentation

#### 5.206.2.1 `shouldRedirectStdOut`

```
bool Catch::ReporterPreferences::shouldRedirectStdOut = false
```

Catch2 should redirect writes to stdout and pass them to the reporter

#### 5.206.2.2 `shouldReportAllAssertions`

```
bool Catch::ReporterPreferences::shouldReportAllAssertions = false
```

Catch2 should call `Reporter::assertionEnded` even for passing assertions

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.207 `Catch::ReporterRegistrar< T >` Class Template Reference

### Public Member Functions

- `ReporterRegistrar` (std::string const &name)

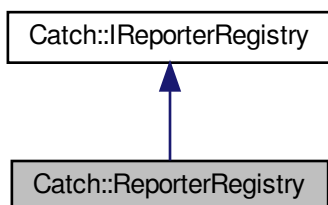
The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

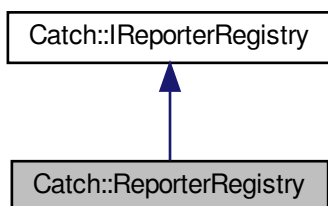


## 5.208 Catch::ReporterRegistry Class Reference

Inheritance diagram for Catch::ReporterRegistry:



Collaboration diagram for Catch::ReporterRegistry:



### Public Member Functions

- [IEventListenerPtr](#) **create** (std::string const &name, [ReporterConfig](#) &&config) const override
- void **registerReporter** (std::string const &name, [IReporterFactoryPtr](#) factory)
- void **registerListener** ([Detail::unique\\_ptr](#)< [EventListenerFactory](#) > factory)
- FactoryMap const & **getFactories** () const override
- Listeners const & **getListeners** () const override

### Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.209 Catch::ReporterSpec Class Reference

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- **ReporterSpec** (std::string name, [Optional](#)< std::string > outputFileName, [Optional](#)< [ColourMode](#) > colourMode, std::map< std::string, std::string > customOptions)
- std::string const & **name** () const
- [Optional](#)< std::string > const & **outputFile** () const
- [Optional](#)< [ColourMode](#) > const & **colourMode** () const
- std::map< std::string, std::string > const & **customOptions** () const

### Friends

- bool **operator==** ([ReporterSpec](#) const &lhs, [ReporterSpec](#) const &rhs)
- bool **operator!=** ([ReporterSpec](#) const &lhs, [ReporterSpec](#) const &rhs)

### 5.209.1 Detailed Description

Structured reporter spec that a reporter can be created from

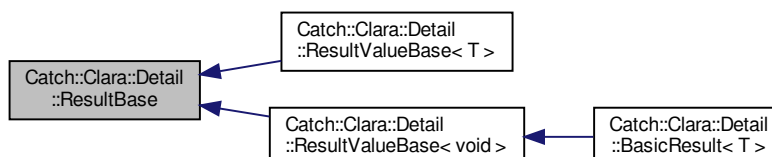
Parsing has been validated, but semantics have not. This means e.g. that the colour mode is known to Catch2, but it might not be compiled into the binary, and the output filename might not be openable.

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.210 Catch::Clara::Detail::ResultBase Class Reference

Inheritance diagram for Catch::Clara::Detail::ResultBase:



## Protected Member Functions

- **ResultBase** ([ResultType](#) type)
- **ResultBase** ([ResultBase](#) const &)=default
- [ResultBase](#) & **operator=** ([ResultBase](#) const &)=default
- **ResultBase** ([ResultBase](#) &&)=default
- [ResultBase](#) & **operator=** ([ResultBase](#) &&)=default
- virtual void **enforceOk** () const =0

## Protected Attributes

- [ResultType](#) m\_type

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.211 Catch::ResultDisposition Struct Reference

### Public Types

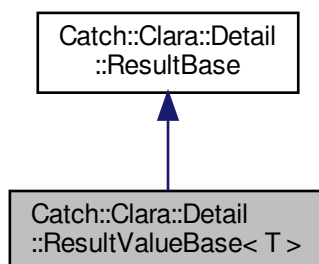
- enum **Flags** { **Normal** = 0x01 , **ContinueOnFailure** = 0x02 , **FalseTest** = 0x04 , **SuppressFail** = 0x08 }

The documentation for this struct was generated from the following file:

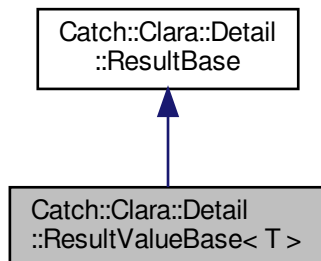
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.212 Catch::Clara::Detail::ResultValueBase< T > Class Template Reference

Inheritance diagram for Catch::Clara::Detail::ResultValueBase< T >:



Collaboration diagram for `Catch::Clara::Detail::ResultValueBase< T >`:



## Public Member Functions

- `auto value () const -> T const &`

## Protected Member Functions

- `ResultValueBase (ResultType type)`
- `ResultValueBase (ResultValueBase const &other)`
- `ResultValueBase (ResultType, T const &value)`
- `auto operator= (ResultValueBase const &other) -> ResultValueBase &`

## Protected Attributes

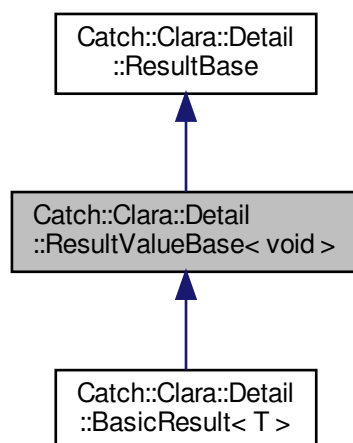
- ```
union {  
    T m_value  
};
```

The documentation for this class was generated from the following file:

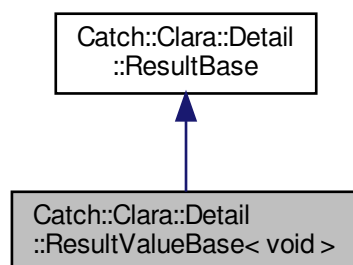
- `include/catch2/catch_amalgamated.hpp`

## 5.213 Catch::Clara::Detail::ResultValueBase< void > Class Reference

Inheritance diagram for Catch::Clara::Detail::ResultValueBase< void >:



Collaboration diagram for Catch::Clara::Detail::ResultValueBase< void >:



### Protected Member Functions

- **ResultBase** ([ResultType](#) type)
- **ResultBase** ([ResultBase](#) const &)=default
- **ResultBase** ([ResultBase](#) &&)=default

### Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.214 Catch::ResultWas Struct Reference

### Public Types

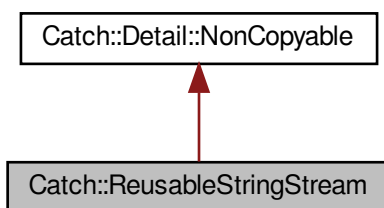
- enum **OfType** {  
    **Unknown** = -1 , **Ok** = 0 , **Info** = 1 , **Warning** = 2 ,  
    **FailureBit** = 0x10 , **ExpressionFailed** = FailureBit | 1 , **ExplicitFailure** = FailureBit | 2 , **Exception** = 0x100  
    | FailureBit ,  
    **ThrewException** = Exception | 1 , **DidntThrowException** = Exception | 2 , **FatalErrorCondition** = 0x200 |  
    FailureBit }

The documentation for this struct was generated from the following file:

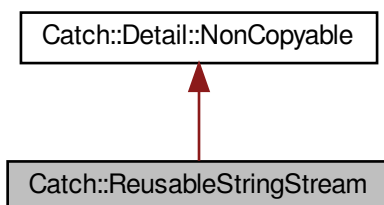
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.215 Catch::ReusableStringStream Class Reference

Inheritance diagram for Catch::ReusableStringStream:



Collaboration diagram for Catch::ReusableStringStream:



## Public Member Functions

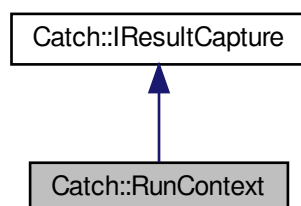
- `std::string str () const`  
*Returns the serialized state.*
- `void str (std::string const &str)`  
*Sets internal state to `str`*
- `template<typename T >`  
`auto operator<< (T const &value) -> ReusableStringStream &`
- `auto get () -> std::ostream &`

The documentation for this class was generated from the following file:

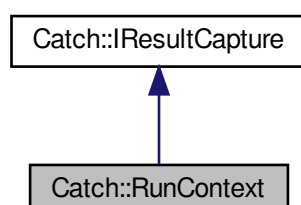
- `include/catch2/catch\_amalgamated.hpp`

## 5.216 Catch::RunContext Class Reference

Inheritance diagram for Catch::RunContext:



Collaboration diagram for Catch::RunContext:



## Public Member Functions

- **RunContext** ([RunContext](#) const &)=delete
- **RunContext** & **operator=** ([RunContext](#) const &)=delete
- **RunContext** ([IConfig](#) const \*\_config, [IEventListenerPtr](#) &&reporter)
- **Totals** **runTest** ([TestCaseHandle](#) const &testCase)
- void **handleExpr** ([AssertionInfo](#) const &info, [ITransientExpression](#) const &expr, [AssertionReaction](#) &reaction) override
- void **handleMessage** ([AssertionInfo](#) const &info, [ResultWas::OfType](#) resultType, [StringRef](#) message, [AssertionReaction](#) &reaction) override
- void **handleUnexpectedExceptionNotThrown** ([AssertionInfo](#) const &info, [AssertionReaction](#) &reaction) override
- void **handleUnexpectedInflightException** ([AssertionInfo](#) const &info, std::string const &message, [AssertionReaction](#) &reaction) override
- void **handleIncomplete** ([AssertionInfo](#) const &info) override
- void **handleNonExpr** ([AssertionInfo](#) const &info, [ResultWas::OfType](#) resultType, [AssertionReaction](#) &reaction) override
- bool **sectionStarted** ([SectionInfo](#) const &sectionInfo, [Counts](#) &assertions) override
- void **sectionEnded** ([SectionEndInfo](#) const &endInfo) override
- void **sectionEndedEarly** ([SectionEndInfo](#) const &endInfo) override
- auto **acquireGeneratorTracker** ([StringRef](#) generatorName, [SourceLineInfo](#) const &lineInfo) -> [IGeneratorTracker](#) &override
- void **benchmarkPreparing** ([StringRef](#) name) override
- void **benchmarkStarting** ([BenchmarkInfo](#) const &info) override
- void **benchmarkEnded** ([BenchmarkStats](#)<> const &stats) override
- void **benchmarkFailed** ([StringRef](#) error) override
- void **pushScopedMessage** ([MessageInfo](#) const &message) override
- void **popScopedMessage** ([MessageInfo](#) const &message) override
- void **emplaceUnscopedMessage** ([MessageBuilder](#) const &builder) override
- std::string **getCurrentTestName** () const override
- const [AssertionResult](#) \* **getLastResult** () const override
- void **exceptionEarlyReported** () override
- void **handleFatalErrorCondition** ([StringRef](#) message) override
- bool **lastAssertionPassed** () override
- void **assertionPassed** () override
- bool **aborting** () const

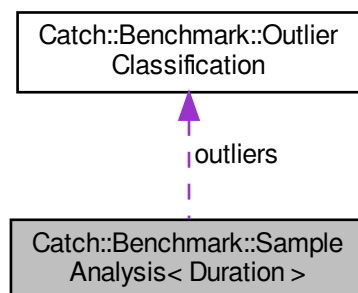
The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



## 5.217 Catch::Benchmark::SampleAnalysis< Duration > Struct Template Reference

Collaboration diagram for Catch::Benchmark::SampleAnalysis< Duration >:



### Public Member Functions

- `template<typename Duration2 > operator SampleAnalysis< Duration2 > () const`

### Public Attributes

- `std::vector< Duration > samples`
- `Estimate< Duration > mean`
- `Estimate< Duration > standard_deviation`
- `OutlierClassification outliers`
- `double outlier_variance`

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.218 Catch::XmlWriter::ScopedElement Class Reference

### Public Member Functions

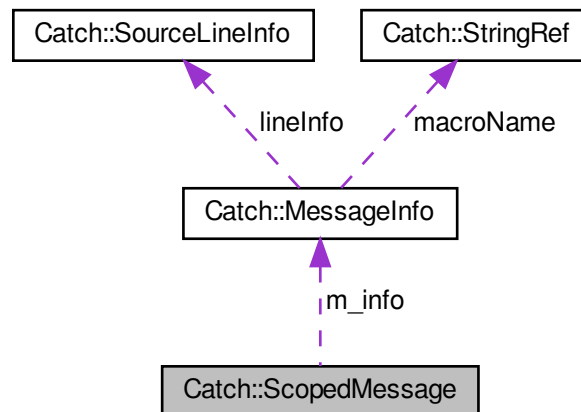
- `ScopedElement (XmlWriter *writer, XmlFormatting fmt)`
- `ScopedElement (ScopedElement &&other) noexcept`
- `ScopedElement & operator= (ScopedElement &&other) noexcept`
- `ScopedElement & writeText (StringRef text, XmlFormatting fmt=XmlFormatting::Newline|XmlFormatting::Indent)`
- `ScopedElement & writeAttribute (StringRef name, StringRef attribute)`
- `template<typename T, typename = typename std::enable_if_t< !std::is_convertible<T, StringRef>::value>> ScopedElement & writeAttribute (StringRef name, T const &attribute)`

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.219 Catch::ScopedMessage Class Reference

Collaboration diagram for Catch::ScopedMessage:



### Public Member Functions

- **ScopedMessage** ([MessageBuilder](#) const &builder)
- **ScopedMessage** ([ScopedMessage](#) &duplicate)=delete
- **ScopedMessage** ([ScopedMessage](#) &&old) noexcept

### Public Attributes

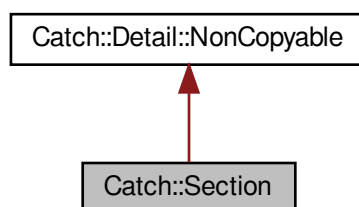
- [MessageInfo](#) **m\_info**
- bool **m\_moved** = false

The documentation for this class was generated from the following file:

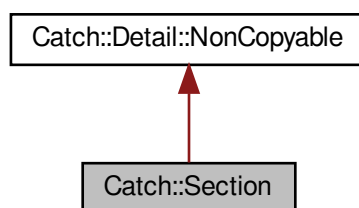
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.220 Catch::Section Class Reference

Inheritance diagram for Catch::Section:



Collaboration diagram for Catch::Section:



### Public Member Functions

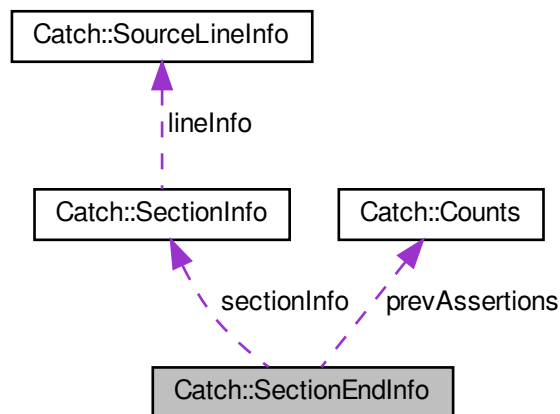
- **Section** ([SectionInfo](#) &&info)
- **operator bool** () const

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.221 Catch::SectionEndInfo Struct Reference

Collaboration diagram for Catch::SectionEndInfo:



### Public Attributes

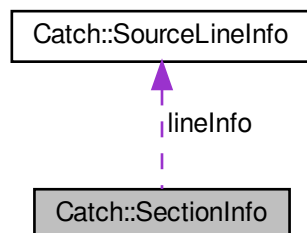
- [SectionInfo](#) `sectionInfo`
- [Counts](#) `prevAssertions`
- double `durationInSeconds`

The documentation for this struct was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.222 Catch::SectionInfo Struct Reference

Collaboration diagram for Catch::SectionInfo:



## Public Member Functions

- **SectionInfo** ([SourceLineInfo](#) const &\_lineInfo, std::string \_name, const char \*const = nullptr)

## Public Attributes

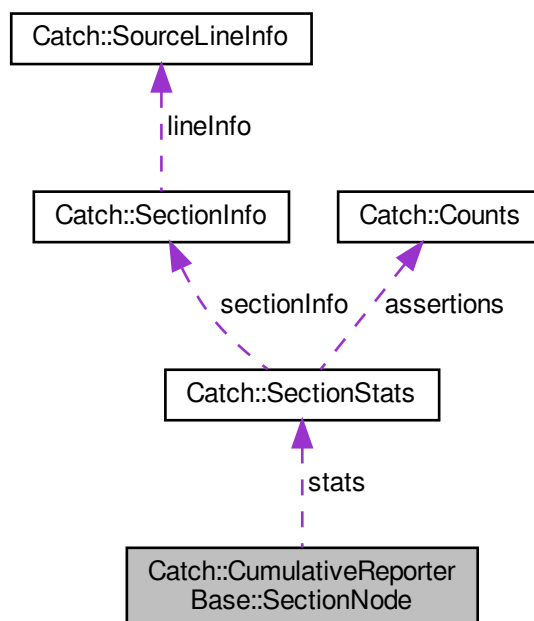
- std::string **name**
- [SourceLineInfo](#) **lineInfo**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.223 Catch::CumulativeReporterBase::SectionNode Struct Reference

Collaboration diagram for Catch::CumulativeReporterBase::SectionNode:



## Public Member Functions

- **SectionNode** ([SectionStats](#) const &\_stats)
- bool **operator==** ([SectionNode](#) const &other) const
- bool **hasAnyAssertions** () const

## Public Attributes

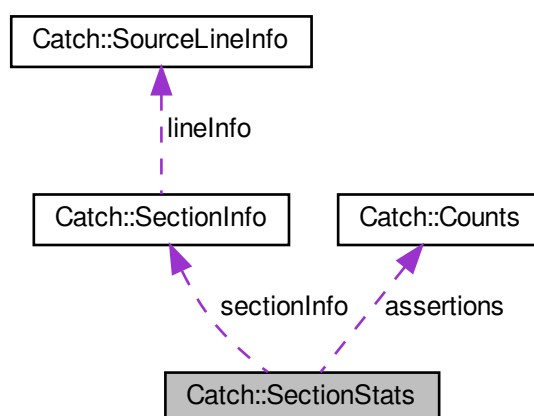
- [SectionStats](#) **stats**
- `std::vector< Detail::unique\_ptr< SectionNode > >` **childSections**
- `std::vector< Detail::AssertionOrBenchmarkResult >` **assertionsAndBenchmarks**
- `std::string` **stdOut**
- `std::string` **stdErr**

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.224 Catch::SectionStats Struct Reference

Collaboration diagram for Catch::SectionStats:



## Public Member Functions

- **SectionStats** ([SectionInfo](#) const &\_sectionInfo, [Counts](#) const &\_assertions, double \_durationInSeconds, bool \_missingAssertions)

## Public Attributes

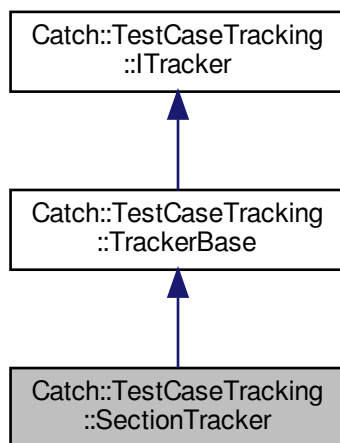
- [SectionInfo](#) **sectionInfo**
- [Counts](#) **assertions**
- double **durationInSeconds**
- bool **missingAssertions**

The documentation for this struct was generated from the following file:

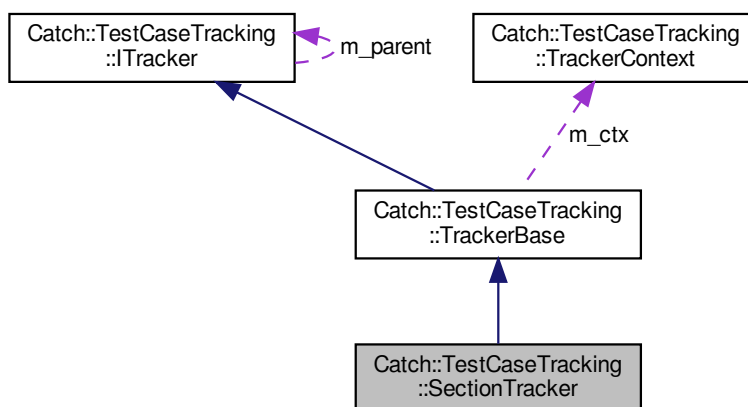
- `include/catch2/catch_amalgamated.hpp`

## 5.225 Catch::TestCaseTracking::SectionTracker Class Reference

Inheritance diagram for Catch::TestCaseTracking::SectionTracker:



Collaboration diagram for Catch::TestCaseTracking::SectionTracker:



### Public Member Functions

- **SectionTracker** ([NameAndLocation](#) const &nameAndLocation, [TrackerContext](#) &ctx, [ITracker](#) \*parent)
  - bool [isSectionTracker](#) () const override
  - bool [isComplete](#) () const override
- Returns true if tracker run to completion (successfully or not)*

- void **tryOpen** ()
- void **addInitialFilters** (std::vector< std::string > const &filters)
- void **addNextFilters** (std::vector< [StringRef](#) > const &filters)
- std::vector< [StringRef](#) > const & **getFilters** () const

*Returns filters active in this tracker.*

- [StringRef](#) **trimmedName** () const

*Returns whitespace-trimmed name of the tracked section.*

## Static Public Member Functions

- static [SectionTracker](#) & **acquire** ([TrackerContext](#) &ctx, [NameAndLocation](#) const &nameAndLocation)

## Additional Inherited Members

### 5.225.1 Member Function Documentation

#### 5.225.1.1 isSectionTracker()

```
bool Catch::TestCaseTracking::SectionTracker::isSectionTracker ( ) const [override], [virtual]
```

Returns true if the instance is a section tracker

Subclasses should override to true if they are, replaces RTTI for internal debug checks.

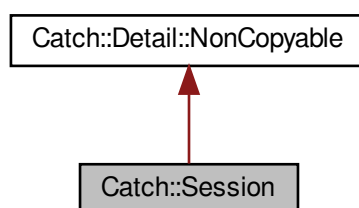
Reimplemented from [Catch::TestCaseTracking::ITracker](#).

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

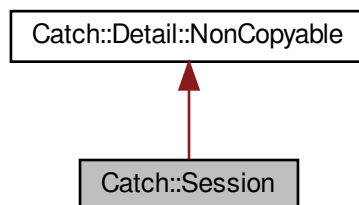
## 5.226 Catch::Session Class Reference

Inheritance diagram for Catch::Session:





Collaboration diagram for Catch::Session:



## Public Member Functions

- void **showHelp** () const
- void **libIdentify** ()
- int **applyCommandLine** (int argc, char const \*const \*argv)
- void **useConfigData** ([ConfigData](#) const &configData)
- template<typename CharT >  
int **run** (int argc, CharT const \*const argv[])
- int **run** ()
- [Clara::Parser](#) const & **cli** () const
- void **cli** ([Clara::Parser](#) const &newParser)
- [ConfigData](#) & **configData** ()
- [Config](#) & **config** ()

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.227 Catch::SimplePcg32 Class Reference

### Public Types

- using **result\_type** = std::uint32\_t

### Public Member Functions

- **SimplePcg32** (result\_type seed\_)
- void **seed** (result\_type seed\_)
- void **discard** (uint64\_t skip)
- result\_type **operator()** ()

## Static Public Member Functions

- static constexpr result\_type() **min** ()
- static constexpr result\_type() **max** ()

## Friends

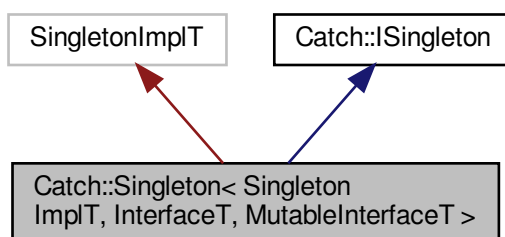
- bool **operator==** (SimplePcg32 const &lhs, SimplePcg32 const &rhs)
- bool **operator!=** (SimplePcg32 const &lhs, SimplePcg32 const &rhs)

The documentation for this class was generated from the following file:

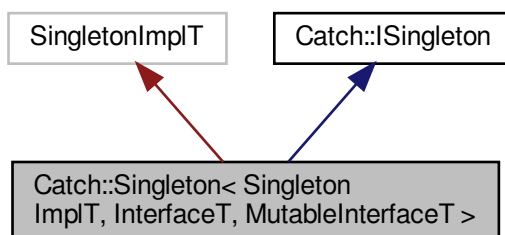
- include/catch2/catch\_amalgamated.hpp

## 5.228 Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT > Class Template Reference

Inheritance diagram for Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT >:



Collaboration diagram for Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT >:



## Static Public Member Functions

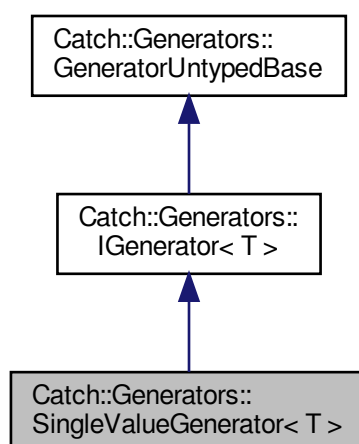
- static auto **get** () -> InterfaceT const &
- static auto **getMutable** () -> MutableInterfaceT &

The documentation for this class was generated from the following file:

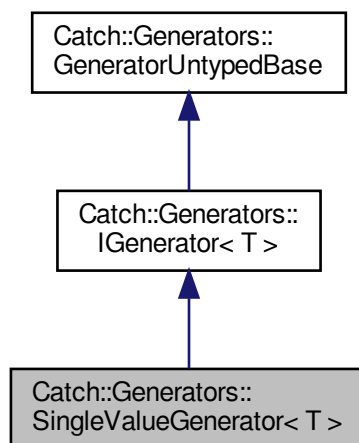
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.229 Catch::Generators::SingleValueGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::SingleValueGenerator< T >:



Collaboration diagram for `Catch::Generators::SingleValueGenerator< T >`:



## Public Member Functions

- **SingleValueGenerator** (T const &value)
- **SingleValueGenerator** (T &&value)
- T const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.229.1 Member Function Documentation

#### 5.229.1.1 next()

```

template<typename T >
bool Catch::Generators::SingleValueGenerator< T >::next ( ) [inline], [override], [virtual]

```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

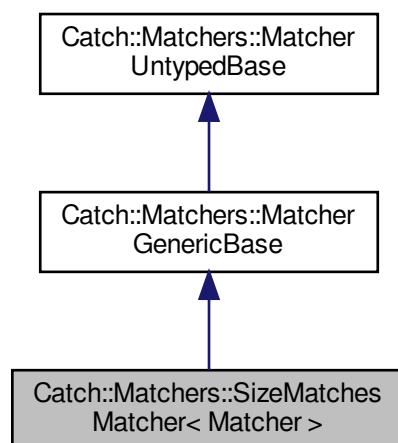
Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

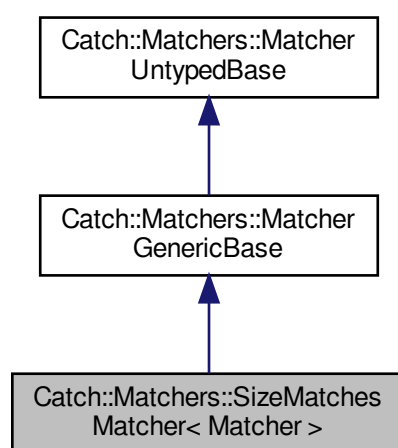
- include/catch2/catch\_amalgamated.hpp

## 5.230 Catch::Matchers::SizeMatchesMatcher< Matcher > Class Template Reference

Inheritance diagram for Catch::Matchers::SizeMatchesMatcher< Matcher >:



Collaboration diagram for Catch::Matchers::SizeMatchesMatcher< Matcher >:



### Public Member Functions

- **SizeMatchesMatcher** (Matcher m)
- `template<typename RangeLike >`  
`bool match (RangeLike &&rng) const`
- `std::string describe () const` override



## Static Public Member Functions

- static std::string **getDescription** ()

## Additional Inherited Members

### 5.231.1 Member Function Documentation

#### 5.231.1.1 testRunStarting()

```
void Catch::SonarQubeReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Reimplemented from [Catch::CumulativeReporterBase](#).

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.232 Catch::SourceLineInfo Struct Reference

### Public Member Functions

- constexpr **SourceLineInfo** (char const \*\_file, std::size\_t \_line) noexcept
- bool **operator==** ([SourceLineInfo](#) const &other) const noexcept
- bool **operator<** ([SourceLineInfo](#) const &other) const noexcept

### Public Attributes

- char const \* **file**
- std::size\_t **line**

### Friends

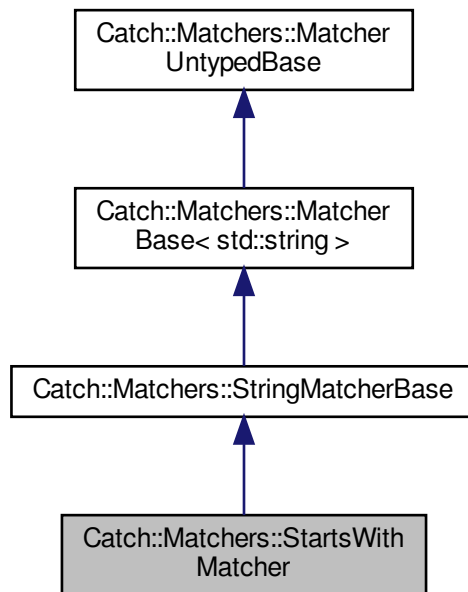
- std::ostream & **operator<<** (std::ostream &os, [SourceLineInfo](#) const &info)

The documentation for this struct was generated from the following file:

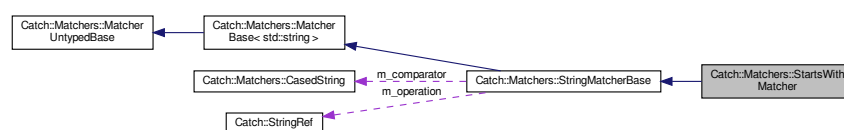
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.233 Catch::Matchers::StartsWithMatcher Class Reference

Inheritance diagram for Catch::Matchers::StartsWithMatcher:



Collaboration diagram for Catch::Matchers::StartsWithMatcher:



### Public Member Functions

- **StartsWithMatcher** ([CasedString](#) const &comparator)
- **bool match** (std::string const &source) const override

### Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



## 5.234 Catch::StartupExceptionRegistry Class Reference

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.235 Catch::StreamEndStop Struct Reference

### Public Member Functions

- [StringRef](#) **operator+** () const

### Friends

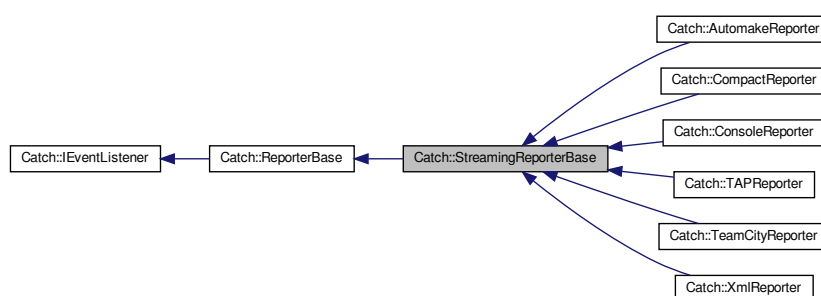
- `template<typename T >`  
`T const & operator+ (T const &value, StreamEndStop)`

The documentation for this struct was generated from the following file:

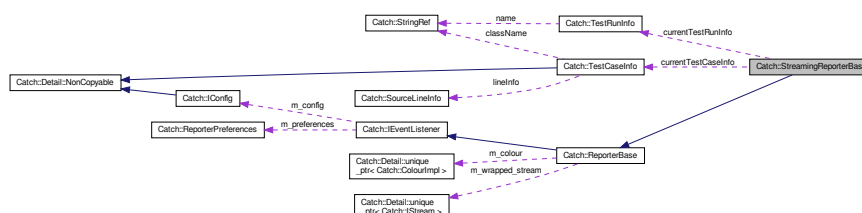
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.236 Catch::StreamingReporterBase Class Reference

Inheritance diagram for Catch::StreamingReporterBase:



Collaboration diagram for Catch::StreamingReporterBase:



## Public Member Functions

- void [benchmarkPreparing](#) ([StringRef](#)) override  
*Called when user-code is being probed before the actual benchmark runs.*
- void [benchmarkStarting](#) ([BenchmarkInfo](#) const &) override  
*Called after probe but before the user-code is being benchmarked.*
- void [benchmarkEnded](#) ([BenchmarkStats](#)<> const &) override  
*Called with the benchmark results if benchmark successfully finishes.*
- void [benchmarkFailed](#) ([StringRef](#)) override  
*Called if running the benchmarks fails for any reason.*
- void [fatalErrorEncountered](#) ([StringRef](#)) override  
*Called if a fatal error (signal/structured exception) occurred.*
- void [noMatchingTestCases](#) ([StringRef](#)) override  
*Called when no test cases match provided test spec.*
- void [reportInvalidTestSpec](#) ([StringRef](#)) override  
*Called for all invalid test specs from the cli.*
- void [testRunStarting](#) ([TestRunInfo](#) const & \_testRunInfo) override
- void [testCaseStarting](#) ([TestCaseInfo](#) const & \_testInfo) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testCasePartialStarting](#) ([TestCaseInfo](#) const &, uint64\_t) override  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- void [sectionStarting](#) ([SectionInfo](#) const & \_sectionInfo) override  
*Called when a SECTION is being entered. Not called for skipped sections.*
- void [assertionStarting](#) ([AssertionInfo](#) const &) override  
*Called before assertion success/failure is evaluated.*
- void [assertionEnded](#) ([AssertionStats](#) const &) override  
*Called after assertion was fully evaluated.*
- void [sectionEnded](#) ([SectionStats](#) const &) override  
*Called after a SECTION has finished running.*
- void [testCasePartialEnded](#) ([TestCaseStats](#) const &, uint64\_t) override  
*Called every time a TEST\_CASE is entered, including repeats (due to sections)*
- void [testCaseEnded](#) ([TestCaseStats](#) const &) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testRunEnded](#) ([TestRunStats](#) const &) override
- void [skipTest](#) ([TestCaseInfo](#) const &) override  
*Called with test cases that are skipped due to the test run aborting.*
- [ReporterBase](#) ([ReporterConfig](#) &&config)

## Protected Attributes

- [TestRunInfo](#) [currentTestRunInfo](#) { "test run has not started yet" \_sr }
- [TestCaseInfo](#) const \* [currentTestCaseInfo](#) = nullptr
- std::vector< [SectionInfo](#) > [m\\_sectionStack](#)  
*Stack of all active sections in the current test case.*

### 5.236.1 Member Function Documentation

### 5.236.1.1 testRunEnded()

```
void Catch::StreamingReporterBase::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Implements [Catch::IEventListener](#).

Reimplemented in [Catch::XmlReporter](#), [Catch::TeamCityReporter](#), [Catch::TAPReporter](#), [Catch::ConsoleReporter](#), and [Catch::CompactReporter](#).

### 5.236.1.2 testRunStarting()

```
void Catch::StreamingReporterBase::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Implements [Catch::IEventListener](#).

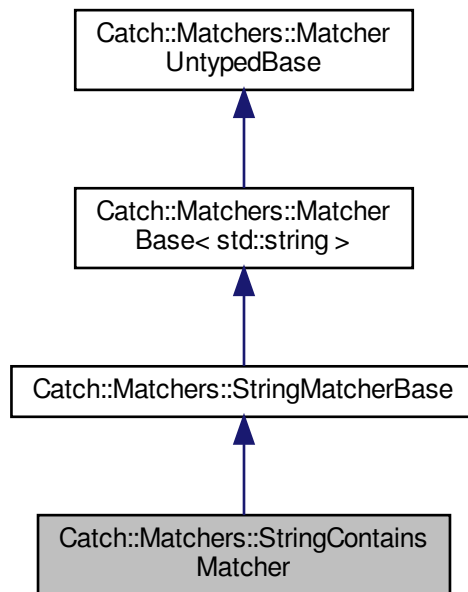
Reimplemented in [Catch::XmlReporter](#), [Catch::TAPReporter](#), [Catch::TeamCityReporter](#), [Catch::ConsoleReporter](#), and [Catch::CompactReporter](#).

The documentation for this class was generated from the following file:

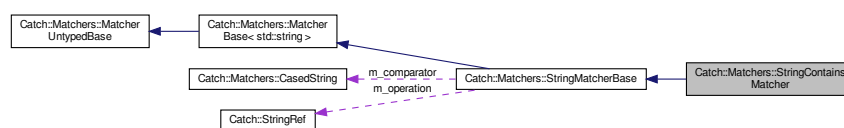
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.237 Catch::Matchers::StringContainsMatcher Class Reference

Inheritance diagram for Catch::Matchers::StringContainsMatcher:



Collaboration diagram for Catch::Matchers::StringContainsMatcher:



### Public Member Functions

- **StringContainsMatcher** ([CasedString](#) const &comparator)
- bool **match** (std::string const &source) const override

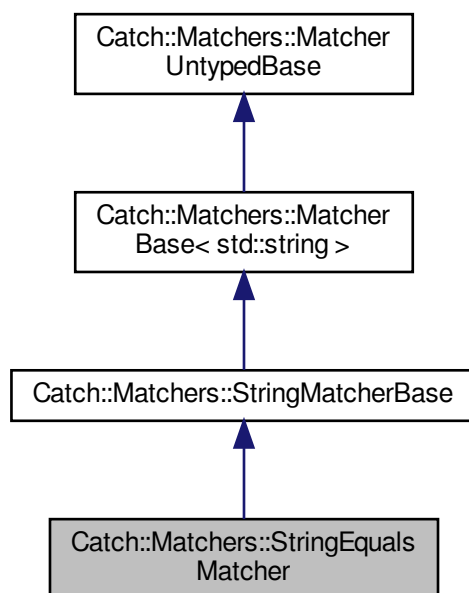
### Additional Inherited Members

The documentation for this class was generated from the following file:

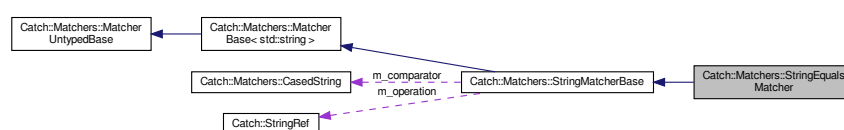
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.238 Catch::Matchers::StringEqualsMatcher Class Reference

Inheritance diagram for Catch::Matchers::StringEqualsMatcher:



Collaboration diagram for Catch::Matchers::StringEqualsMatcher:



### Public Member Functions

- **StringEqualsMatcher** ([CasedString](#) const &comparator)
- bool **match** (std::string const &source) const override

### Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.239 Catch::StringMaker< T, typename > Struct Template Reference

### Static Public Member Functions

- `template<typename Fake = T>`  
`static std::enable_if_t<!: Catch::Detail::IsStreamInsertable< Fake >::value, std::string > convert (const Fake &value)`
- `template<typename Fake = T>`  
`static std::enable_if_t<!: Catch::Detail::IsStreamInsertable< Fake >::value, std::string > convert (const Fake &value)`

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.240 Catch::StringMaker< bool > Struct Reference

### Static Public Member Functions

- `static std::string convert (bool b)`

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.241 Catch::StringMaker< Catch::Approx > Struct Reference

### Static Public Member Functions

- `static std::string convert (Catch::Approx const &value)`

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.242 Catch::StringMaker< char \* > Struct Reference

### Static Public Member Functions

- `static std::string convert (char *str)`

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.243 Catch::StringMaker< char > Struct Reference

### Static Public Member Functions

- static std::string **convert** (char c)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.244 Catch::StringMaker< char const \* > Struct Reference

### Static Public Member Functions

- static std::string **convert** (char const \*str)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.245 Catch::StringMaker< char[SZ]> Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (char const \*str)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.246 Catch::StringMaker< double > Struct Reference

### Static Public Member Functions

- static std::string **convert** (double value)

### Static Public Attributes

- static int **precision**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.247 Catch::StringMaker< float > Struct Reference

### Static Public Member Functions

- static std::string **convert** (float value)

### Static Public Attributes

- static int **precision**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.248 Catch::StringMaker< int > Struct Reference

### Static Public Member Functions

- static std::string **convert** (int value)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.249 Catch::StringMaker< long > Struct Reference

### Static Public Member Functions

- static std::string **convert** (long value)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.250 Catch::StringMaker< long long > Struct Reference

### Static Public Member Functions

- static std::string **convert** (long long value)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



## 5.251 Catch::StringMaker< R C::\* > Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (R C::\*p)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.252 Catch::StringMaker< R, std::enable\_if\_t< is\_range< R >::value &&!::Catch::Detail::IsStreamInsertable< R >::value > > Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (R const &range)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.253 Catch::StringMaker< signed char > Struct Reference

### Static Public Member Functions

- static std::string **convert** (signed char c)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.254 Catch::StringMaker< signed char[SZ]> Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (signed char const \*str)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.255 **Catch::StringMaker**< **std::chrono::duration**< **Value**, **Ratio** > > **Struct Template Reference**

### Static Public Member Functions

- static std::string **convert** (std::chrono::duration< Value, Ratio > const &duration)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.256 **Catch::StringMaker**< **std::chrono::duration**< **Value**, **std::ratio**< 1 > > > **Struct Template Reference**

### Static Public Member Functions

- static std::string **convert** (std::chrono::duration< Value, std::ratio< 1 > > const &duration)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.257 **Catch::StringMaker**< **std::chrono::duration**< **Value**, **std::ratio**< **3600** > > > **Struct Template Reference**

### Static Public Member Functions

- static std::string **convert** (std::chrono::duration< Value, std::ratio< 3600 > > const &duration)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.258 **Catch::StringMaker**< **std::chrono::duration**< **Value**, **std::ratio**< 60 > > > **Struct Template Reference**

### Static Public Member Functions

- static std::string **convert** (std::chrono::duration< Value, std::ratio< 60 > > const &duration)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.259 Catch::StringMaker< std::chrono::time\_point< Clock, Duration > > Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (std::chrono::time\_point< Clock, Duration > const &time\_point)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.260 Catch::StringMaker< std::chrono::time\_point< std::chrono::system\_clock, Duration > > Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (std::chrono::time\_point< std::chrono::system\_clock, Duration > const &time\_point)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.261 Catch::StringMaker< std::nullptr\_t > Struct Reference

### Static Public Member Functions

- static std::string **convert** (std::nullptr\_t)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.262 Catch::StringMaker< std::string > Struct Reference

### Static Public Member Functions

- static std::string **convert** (const std::string &str)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.263 Catch::StringMaker< std::wstring > Struct Reference

### Static Public Member Functions

- static std::string **convert** (const std::wstring &wstr)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.264 Catch::StringMaker< T \* > Struct Template Reference

### Static Public Member Functions

- template<typename U >  
static std::string **convert** (U \*p)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.265 Catch::StringMaker< T[SZ]> Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (T const(&arr)[SZ])

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.266 Catch::StringMaker< unsigned char > Struct Reference

### Static Public Member Functions

- static std::string **convert** (unsigned char c)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.267 Catch::StringMaker< unsigned char[SZ]> Struct Template Reference

### Static Public Member Functions

- static std::string **convert** (unsigned char const \*str)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.268 Catch::StringMaker< unsigned int > Struct Reference

### Static Public Member Functions

- static std::string **convert** (unsigned int value)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.269 Catch::StringMaker< unsigned long > Struct Reference

### Static Public Member Functions

- static std::string **convert** (unsigned long value)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.270 Catch::StringMaker< unsigned long long > Struct Reference

### Static Public Member Functions

- static std::string **convert** (unsigned long long value)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.271 Catch::StringMaker< wchar\_t \* > Struct Reference

### Static Public Member Functions

- static std::string **convert** (wchar\_t \*str)

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.272 Catch::StringMaker< wchar\_t const \* > Struct Reference

### Static Public Member Functions

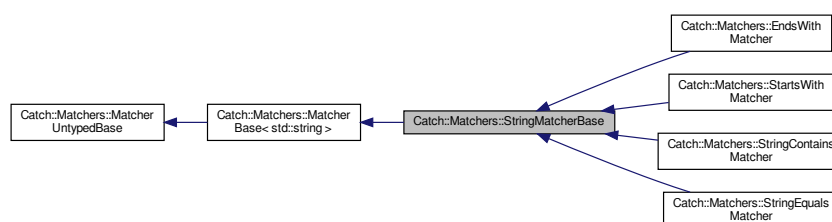
- static std::string **convert** (wchar\_t const \*str)

The documentation for this struct was generated from the following file:

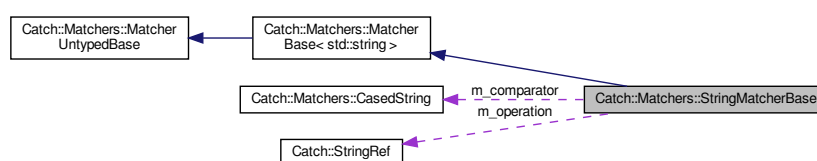
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.273 Catch::Matchers::StringMatcherBase Class Reference

Inheritance diagram for Catch::Matchers::StringMatcherBase:



Collaboration diagram for Catch::Matchers::StringMatcherBase:



## Public Member Functions

- **StringMatcherBase** ([StringRef](#) operation, [CasedString](#) const &comparator)
- `std::string describe ()` const override

## Protected Attributes

- [CasedString](#) m\_comparator
- [StringRef](#) m\_operation

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.274 Catch::StringRef Class Reference

```
#include <catch_amalgamated.hpp>
```

## Public Types

- using **size\_type** = std::size\_t
- using **const\_iterator** = const char \*

## Public Member Functions

- **StringRef** (char const \*rawChars) noexcept
- constexpr **StringRef** (char const \*rawChars, size\_type size) noexcept
- **StringRef** (std::string const &stdString) noexcept
- **operator std::string ()** const
- auto **operator==** ([StringRef](#) other) const noexcept -> bool
- auto **operator!=** ([StringRef](#) other) const noexcept -> bool
- constexpr auto **operator[]** (size\_type index) const noexcept -> char
- bool **operator<** ([StringRef](#) rhs) const noexcept
- constexpr auto **empty** () const noexcept -> bool
- constexpr auto **size** () const noexcept -> size\_type
- constexpr [StringRef](#) **substr** (size\_type start, size\_type length) const noexcept
- constexpr char const \* **data** () const noexcept
- constexpr const\_iterator **begin** () const
- constexpr const\_iterator **end** () const
- int **compare** ([StringRef](#) rhs) const

## Friends

- std::string & **operator+=** (std::string &lhs, [StringRef](#) sr)
- std::ostream & **operator<<** (std::ostream &os, [StringRef](#) sr)
- std::string **operator+** ([StringRef](#) lhs, [StringRef](#) rhs)

### 5.274.1 Detailed Description

A non-owning string class (similar to the forthcoming `std::string_view`) Note that, because a [StringRef](#) may be a substring of another string, it may not be null terminated.

### 5.274.2 Member Function Documentation

#### 5.274.2.1 `compare()`

```
int Catch::StringRef::compare (
    StringRef rhs ) const
```

Provides a three-way comparison with rhs

Returns negative number if lhs < rhs, 0 if lhs == rhs, and a positive number if lhs > rhs

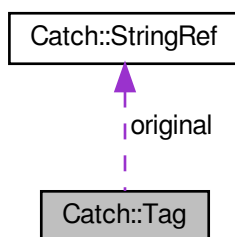
The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.275 Catch::Tag Struct Reference

```
#include <catch_amalgamated.hpp>
```

Collaboration diagram for `Catch::Tag`:



### Public Member Functions

- constexpr **Tag** ([StringRef](#) original\_)



## Public Attributes

- [StringRef](#) **original**

## Friends

- bool **operator**< ([Tag](#) const &lhs, [Tag](#) const &rhs)
- bool **operator**== ([Tag](#) const &lhs, [Tag](#) const &rhs)

### 5.275.1 Detailed Description

A **view** of a tag string that provides case insensitive comparisons

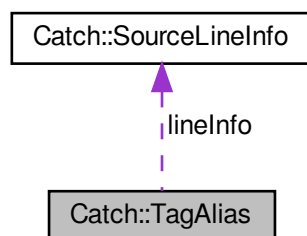
Note that in Catch2 internals, the square brackets around tags are not a part of tag's representation, so e.g. "[cool-tag]" is represented as "cool-tag" internally.

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.276 Catch::TagAlias Struct Reference

Collaboration diagram for Catch::TagAlias:



## Public Member Functions

- **TagAlias** (std::string const &\_tag, [SourceLineInfo](#) \_lineInfo)

## Public Attributes

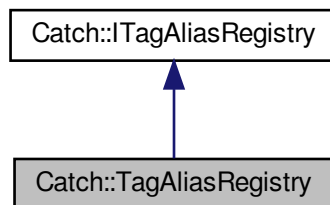
- std::string **tag**
- [SourceLineInfo](#) **lineInfo**

The documentation for this struct was generated from the following file:

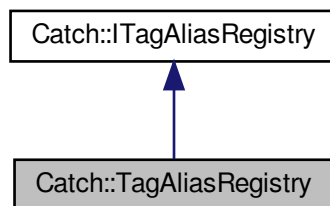
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.277 Catch::TagAliasRegistry Class Reference

Inheritance diagram for Catch::TagAliasRegistry:



Collaboration diagram for Catch::TagAliasRegistry:



### Public Member Functions

- [TagAlias](#) const \* **find** (std::string const &alias) const override
- std::string **expandAliases** (std::string const &unexpandedTestSpec) const override
- void **add** (std::string const &alias, std::string const &tag, [SourceLineInfo](#) const &lineInfo)

### Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.278 Catch::TagInfo Struct Reference

### Public Member Functions

- void **add** ([StringRef](#) spelling)
- std::string **all** () const

## Public Attributes

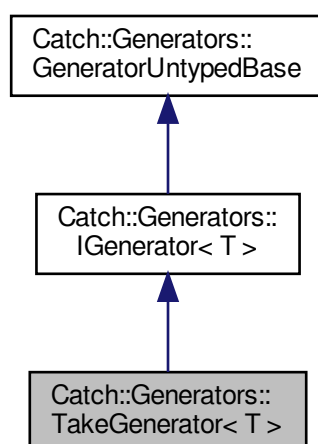
- `std::set< StringRef > spellings`
- `std::size_t count = 0`

The documentation for this struct was generated from the following file:

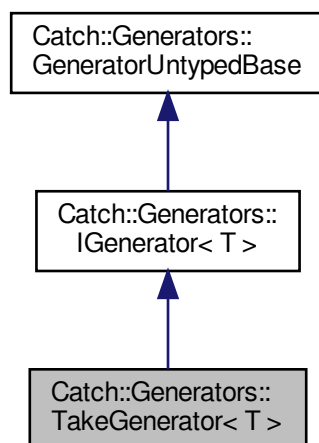
- `include/catch2/catch\_amalgamated.hpp`

## 5.279 Catch::Generators::TakeGenerator< T > Class Template Reference

Inheritance diagram for Catch::Generators::TakeGenerator< T >:



Collaboration diagram for `Catch::Generators::TakeGenerator< T >`:



## Public Member Functions

- **TakeGenerator** (`size_t` target, [GeneratorWrapper< T >](#) &&generator)
- `T` const & **get** () const override
- bool **next** () override

## Additional Inherited Members

### 5.279.1 Member Function Documentation

#### 5.279.1.1 next()

```

template<typename T >
bool Catch::Generators::TakeGenerator< T >::next ( ) [inline], [override], [virtual]
  
```

Attempts to move the generator to the next element

Returns true iff the move succeeded (and a valid element can be retrieved).

Implements [Catch::Generators::GeneratorUntypedBase](#).

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



## Additional Inherited Members

### 5.280.1 Member Function Documentation

#### 5.280.1.1 testRunEnded()

```
void Catch::TAPReporter::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Reimplemented from [Catch::StreamingReporterBase](#).

#### 5.280.1.2 testRunStarting()

```
void Catch::TAPReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

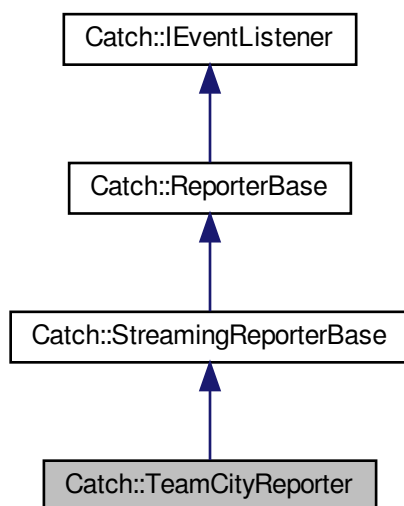
Reimplemented from [Catch::StreamingReporterBase](#).

The documentation for this class was generated from the following file:

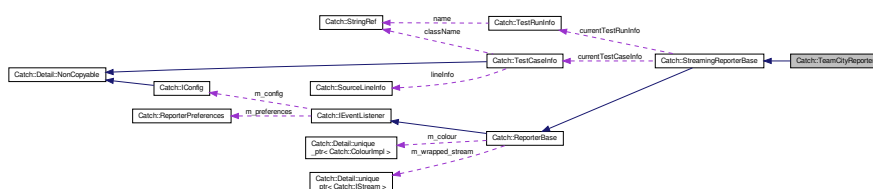
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.281 Catch::TeamCityReporter Class Reference

Inheritance diagram for Catch::TeamCityReporter:



Collaboration diagram for Catch::TeamCityReporter:



## Public Member Functions

- **TeamCityReporter** ([ReporterConfig](#) && [\\_config](#))
- void [testRunStarting](#) ([TestRunInfo](#) const &groupInfo) override
- void [testRunEnded](#) ([TestRunStats](#) const &testGroupStats) override
- void [assertionEnded](#) ([AssertionStats](#) const &assertionStats) override  
*Called after assertion was fully evaluated.*
- void [sectionStarting](#) ([SectionInfo](#) const &sectionInfo) override  
*Called when a SECTION is being entered. Not called for skipped sections.*
- void [testCaseStarting](#) ([TestCaseInfo](#) const &testInfo) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*
- void [testCaseEnded](#) ([TestCaseStats](#) const &testCaseStats) override  
*Called once for each TEST\_CASE, no matter how many times it is entered.*

## Static Public Member Functions

- static std::string **getDescription** ()

## Additional Inherited Members

### 5.281.1 Member Function Documentation

#### 5.281.1.1 testRunEnded()

```
void Catch::TeamCityReporter::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Reimplemented from [Catch::StreamingReporterBase](#).

#### 5.281.1.2 testRunStarting()

```
void Catch::TeamCityReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Reimplemented from [Catch::StreamingReporterBase](#).

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.282 Catch::TestCaseHandle Class Reference

```
#include <catch_amalgamated.hpp>
```

### Public Member Functions

- **TestCaseHandle** ([TestCaseInfo](#) \*info, [ITestInvoker](#) \*invoker)
- void **invoke** () const
- [TestCaseInfo](#) const & **getTestCaseInfo** () const



### 5.282.1 Detailed Description

Wrapper over the test case information and the test case invoker

Does not own either, and is specifically made to be cheap to copy around.

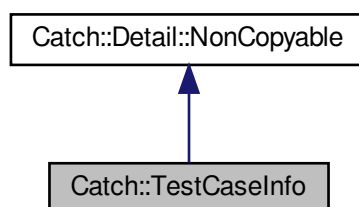
The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

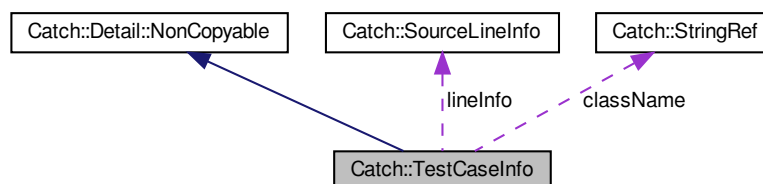
## 5.283 Catch::TestCaseInfo Struct Reference

```
#include <catch_amalgamated.hpp>
```

Inheritance diagram for Catch::TestCaseInfo:



Collaboration diagram for Catch::TestCaseInfo:



### Public Member Functions

- **TestCaseInfo** ([StringRef](#) \_className, [NameAndTags](#) const &\_tags, [SourceLineInfo](#) const &\_lineInfo)
- bool **isHidden** () const
- bool **throws** () const
- bool **okToFail** () const
- bool **expectedToFail** () const
- void **addFilenameTag** ()
- std::string **tagsAsString** () const

## Public Attributes

- `std::string` **name**
- [StringRef](#) **className**
- `std::vector< Tag >` **tags**
- [SourceLineInfo](#) **lineInfo**
- `TestCaseProperties` **properties** = `TestCaseProperties::None`

## Friends

- `bool` [operator<](#) ([TestCaseInfo](#) const &lhs, [TestCaseInfo](#) const &rhs)  
*Orders by name, classname and tags.*

### 5.283.1 Detailed Description

Various metadata about the test case.

A test case is uniquely identified by its (class)name and tags combination, with source location being ignored, and other properties being determined from tags.

Tags are kept sorted.

The documentation for this struct was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.284 Catch::TestCaseInfoHasher Class Reference

### Public Types

- using **hash\_t** = `std::uint64_t`

### Public Member Functions

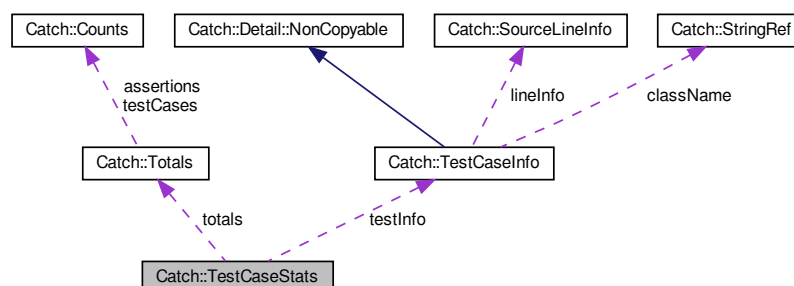
- **TestCaseInfoHasher** (`hash_t` seed)
- `uint32_t` **operator()** ([TestCaseInfo](#) const &t) const

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.285 Catch::TestCaseStats Struct Reference

Collaboration diagram for Catch::TestCaseStats:



### Public Member Functions

- **TestCaseStats** ([TestCaseInfo](#) const &\_testInfo, [Totals](#) const &\_totals, std::string const &\_stdOut, std::string const &\_stdErr, bool \_aborting)

### Public Attributes

- [TestCaseInfo](#) const \* **testInfo**
- [Totals](#) **totals**
- std::string **stdOut**
- std::string **stdErr**
- bool **aborting**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.286 Catch::TestFailureException Struct Reference

Used to signal that an assertion macro failed.

```
#include <catch_amalgamated.hpp>
```

### 5.286.1 Detailed Description

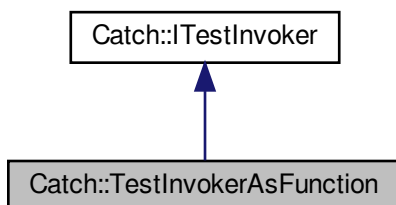
Used to signal that an assertion macro failed.

The documentation for this struct was generated from the following file:

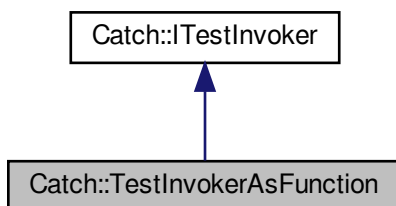
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.287 Catch::TestInvokerAsFunction Class Reference

Inheritance diagram for Catch::TestInvokerAsFunction:



Collaboration diagram for Catch::TestInvokerAsFunction:



### Public Member Functions

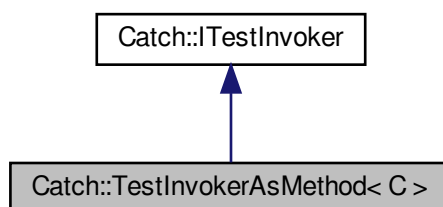
- **TestInvokerAsFunction** (TestType testAsFunction) noexcept
- void **invoke** () const override

The documentation for this class was generated from the following file:

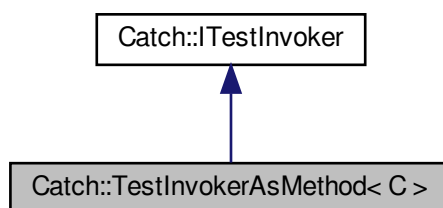
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.288 Catch::TestInvokerAsMethod< C > Class Template Reference

Inheritance diagram for Catch::TestInvokerAsMethod< C >:



Collaboration diagram for Catch::TestInvokerAsMethod< C >:



### Public Member Functions

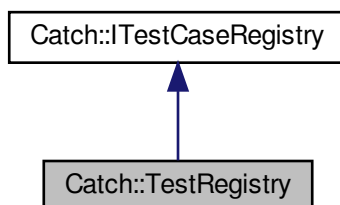
- **TestInvokerAsMethod** (void(C::\*testAsMethod)()) noexcept
- void **invoke** () const override

The documentation for this class was generated from the following file:

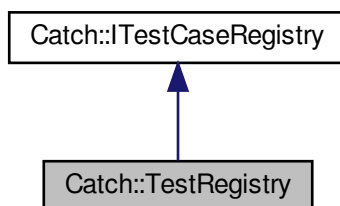
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.289 Catch::TestRegistry Class Reference

Inheritance diagram for Catch::TestRegistry:



Collaboration diagram for Catch::TestRegistry:



### Public Member Functions

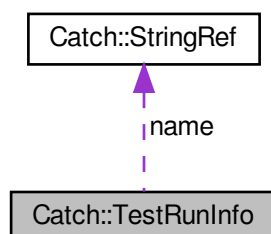
- void **registerTest** ([Detail::unique\\_ptr](#)< [TestCaseInfo](#) > testInfo, [Detail::unique\\_ptr](#)< [ITestInvoker](#) > testInvoker)
- [std::vector](#)< [TestCaseInfo](#) \* > const & **getAllInfos** () const override
- [std::vector](#)< [TestCaseHandle](#) > const & **getAllTests** () const override
- [std::vector](#)< [TestCaseHandle](#) > const & **getAllTestsSorted** ([IConfig](#) const &config) const override

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.290 Catch::TestRunInfo Struct Reference

Collaboration diagram for Catch::TestRunInfo:



### Public Member Functions

- constexpr **TestRunInfo** ([StringRef](#) \_name)

### Public Attributes

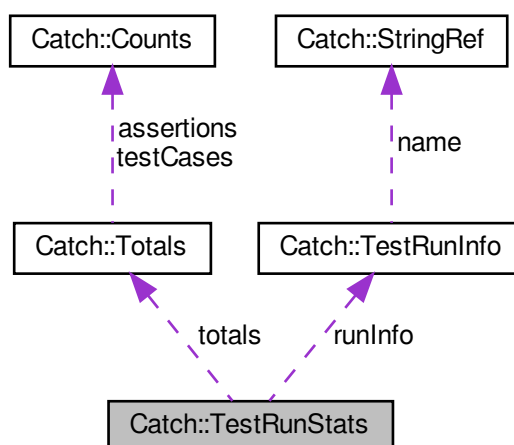
- [StringRef](#) name

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.291 Catch::TestRunStats Struct Reference

Collaboration diagram for Catch::TestRunStats:



## Public Member Functions

- **TestRunStats** ([TestRunInfo](#) const &\_runInfo, [Totals](#) const &\_totals, bool \_aborting)

## Public Attributes

- [TestRunInfo](#) **runInfo**
- [Totals](#) **totals**
- bool **aborting**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.292 Catch::TestSpec Class Reference

### Classes

- struct [FilterMatch](#)

### Public Types

- using **Matches** = std::vector< [FilterMatch](#) >
- using **vectorStrings** = std::vector< std::string >

### Public Member Functions

- bool **hasFilters** () const
- bool **matches** ([TestCaseInfo](#) const &testCase) const
- Matches **matchesByFilter** (std::vector< [TestCaseHandle](#) > const &testCases, [IConfig](#) const &config) const
- const vectorStrings & **getInvalidSpecs** () const

### Friends

- class **TestSpecParser**

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



## 5.293 Catch::TestSpecParser Class Reference

### Public Member Functions

- **TestSpecParser** ([ITagAliasRegistry](#) const &tagAliases)
- **TestSpecParser** & **parse** (std::string const &arg)
- **TestSpec** **testSpec** ()

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.294 Catch::Timer Class Reference

### Public Member Functions

- void **start** ()
- auto **getElapsedNanoseconds** () const -> uint64\_t
- auto **getElapsedMicroseconds** () const -> uint64\_t
- auto **getElapsedMilliseconds** () const -> unsigned int
- auto **getElapsedSeconds** () const -> double

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.295 Catch::Benchmark::Timing< Duration, Result > Struct Template Reference

### Public Attributes

- Duration **elapsed**
- Result **result**
- int **iterations**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.296 Catch::Clara::Detail::Token Struct Reference

### Public Attributes

- TokenType **type**
- std::string **token**

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.297 Catch::Clara::Detail::TokenStream Class Reference

### Public Member Functions

- **TokenStream** ([Args](#) const &args)
- **TokenStream** (Iterator it, Iterator itEnd)
- **operator bool** () const
- **size\_t count** () const
- [Token](#) **operator\*** () const
- [Token](#) const \* **operator->** () const
- [TokenStream](#) & **operator++** ()

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.298 Tool< TSeq > Class Template Reference

[Tools](#) for defending the agent against the virus.

```
#include <tool-bones.hpp>
```

### Public Member Functions

- **Tool** (std::string name="unknown tool")
- void **set\_sequence** (TSeq d)
- void **set\_sequence\_unique** (TSeq d)
- void **set\_sequence** (std::shared\_ptr< TSeq > d)
- std::shared\_ptr< TSeq > **get\_sequence** ()
- TSeq & **get\_sequence\_unique** ()
- void **set\_name** (std::string name)
- std::string **get\_name** () const
- [Agent](#)< TSeq > \* **get\_agent** ()
- int **get\_id** () const
- void **set\_id** (int id)
- void **set\_date** (int d)
- int **get\_date** () const
- void **set\_status** (epiworld\_fast\_int init, epiworld\_fast\_int post)
- void **set\_queue** (epiworld\_fast\_int init, epiworld\_fast\_int post)
- void **get\_status** (epiworld\_fast\_int \*init, epiworld\_fast\_int \*post)
- void **get\_queue** (epiworld\_fast\_int \*init, epiworld\_fast\_int \*post)

### Get and set the tool functions

#### Parameters

|     |                                        |
|-----|----------------------------------------|
| v   | <i>The virus over which to operate</i> |
| fun | <i>the function to be used</i>         |

**Returns***epiworld\_double*

- *epiworld\_double* **get\_susceptibility\_reduction** (VirusPtr< TSeq > v)
- *epiworld\_double* **get\_transmission\_reduction** (VirusPtr< TSeq > v)
- *epiworld\_double* **get\_recovery\_enhancer** (VirusPtr< TSeq > v)
- *epiworld\_double* **get\_death\_reduction** (VirusPtr< TSeq > v)
- void **set\_susceptibility\_reduction\_fun** (ToolFun< TSeq > fun)
- void **set\_transmission\_reduction\_fun** (ToolFun< TSeq > fun)
- void **set\_recovery\_enhancer\_fun** (ToolFun< TSeq > fun)
- void **set\_death\_reduction\_fun** (ToolFun< TSeq > fun)
- void **set\_susceptibility\_reduction** (*epiworld\_double* \*prob)
- void **set\_transmission\_reduction** (*epiworld\_double* \*prob)
- void **set\_recovery\_enhancer** (*epiworld\_double* \*prob)
- void **set\_death\_reduction** (*epiworld\_double* \*prob)
- void **set\_susceptibility\_reduction** (*epiworld\_double* prob)
- void **set\_transmission\_reduction** (*epiworld\_double* prob)
- void **set\_recovery\_enhancer** (*epiworld\_double* prob)
- void **set\_death\_reduction** (*epiworld\_double* prob)

**Friends**

- class **Agent**< TSeq >
- class **Model**< TSeq >
- void **default\_add\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_add\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)

**5.298.1 Detailed Description**

```
template<typename TSeq = int>
class Tool< TSeq >
```

[Tools](#) for defending the agent against the virus.

**Template Parameters**

|             |                  |
|-------------|------------------|
| <i>TSeq</i> | Type of sequence |
|-------------|------------------|

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/tool-bones.hpp
- include/epiworld/tool-meat.hpp

**5.299 Tools< TSeq > Class Template Reference**

Set of tools (useful for building iterators)

```
#include <tools-bones.hpp>
```

## Public Member Functions

- **Tools** ([Agent](#)< TSeq > &p)
- std::vector< TOOLPTR >::iterator **begin** ()
- std::vector< TOOLPTR >::iterator **end** ()
- TOOLPTR & **operator()** (size\_t i)
- TOOLPTR & **operator[]** (size\_t i)
- size\_t **size** () const noexcept

## Friends

- class **Tool**< TSeq >
- class **Agent**< TSeq >

### 5.299.1 Detailed Description

```
template<typename TSeq>
class Tools< TSeq >
```

Set of tools (useful for building iterators)

#### Template Parameters

|             |  |
|-------------|--|
| <i>TSeq</i> |  |
|-------------|--|

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/tools-bones.hpp

## 5.300 Tools\_const< TSeq > Class Template Reference

Set of [Tools](#) (const) (useful for iterators)

```
#include <tools-bones.hpp>
```

## Public Member Functions

- **Tools\_const** (const [Agent](#)< TSeq > &p)
- std::vector< TOOLPTR >::const\_iterator **begin** ()
- std::vector< TOOLPTR >::const\_iterator **end** ()
- const TOOLPTR & **operator()** (size\_t i)
- const TOOLPTR & **operator[]** (size\_t i)
- size\_t **size** () const noexcept

## Friends

- class **Tool**< TSeq >
- class **Agent**< TSeq >

### 5.300.1 Detailed Description

```
template<typename TSeq>
class Tools_const< TSeq >
```

Set of [Tools](#) (const) (useful for iterators)

Template Parameters

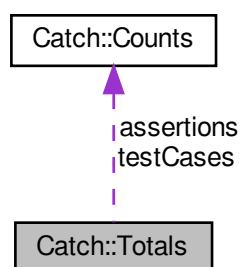
|             |  |
|-------------|--|
| <i>TSeq</i> |  |
|-------------|--|

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/tools-bones.hpp

## 5.301 Catch::Totals Struct Reference

Collaboration diagram for Catch::Totals:



## Public Member Functions

- [Totals](#) **operator-** ([Totals](#) const &other) const
- [Totals](#) & **operator+=** ([Totals](#) const &other)
- [Totals](#) **delta** ([Totals](#) const &prevTotals) const

## Public Attributes

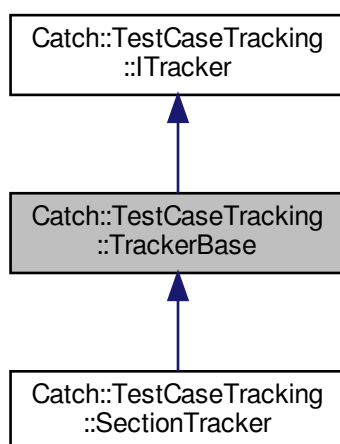
- [Counts](#) assertions
- [Counts](#) testCases

The documentation for this struct was generated from the following file:

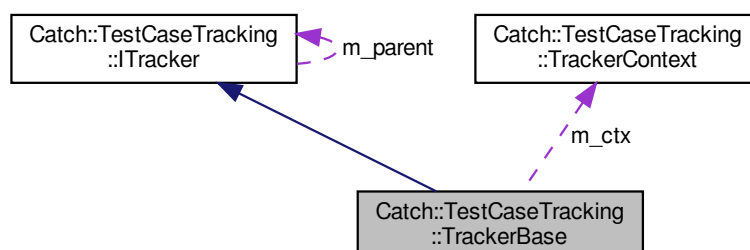
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.302 Catch::TestCaseTracking::TrackerBase Class Reference

Inheritance diagram for Catch::TestCaseTracking::TrackerBase:



Collaboration diagram for Catch::TestCaseTracking::TrackerBase:



## Public Member Functions

- **TrackerBase** ([NameAndLocation](#) const &nameAndLocation, [TrackerContext](#) &ctx, [ITracker](#) \*parent)
- bool [isComplete](#) () const override  
*Returns true if tracker run to completion (successfully or not)*
- void **open** ()
- void **close** () override
- void **fail** () override

## Protected Attributes

- [TrackerContext](#) & m\_ctx

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.303 Catch::TestCaseTracking::TrackerContext Class Reference

### Public Member Functions

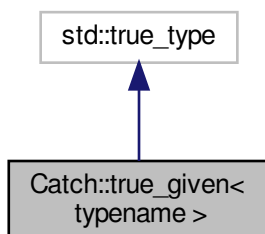
- [ITracker](#) & **startRun** ()
- void **endRun** ()
- void **startCycle** ()
- void **completeCycle** ()
- bool **completedCycle** () const
- [ITracker](#) & **currentTracker** ()
- void **setCurrentTracker** ([ITracker](#) \*tracker)

The documentation for this class was generated from the following file:

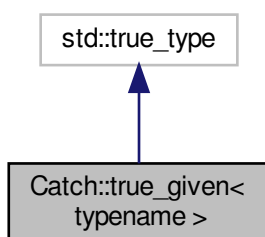
- include/catch2/[catch\\_amalgamated.hpp](#)

### 5.304 Catch::true\_given< typename > Struct Template Reference

Inheritance diagram for Catch::true\_given< typename >:



Collaboration diagram for Catch::true\_given< typename >:



The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

### 5.305 Catch::Benchmark::Detail::CompleteType< void >::type Struct Reference

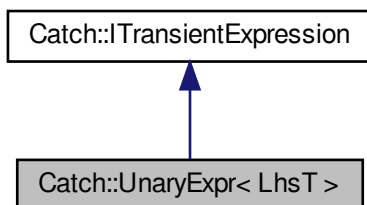
The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

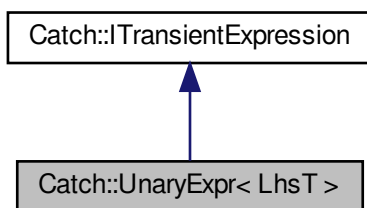


## 5.306 Catch::UnaryExpr< LhsT > Class Template Reference

Inheritance diagram for Catch::UnaryExpr< LhsT >:



Collaboration diagram for Catch::UnaryExpr< LhsT >:



### Public Member Functions

- **UnaryExpr** (LhsT lhs)

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.307 Catch::Clara::Detail::UnaryLambdaTraits< L > Struct Template Reference

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

### 5.308 Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::\*)(Args...) const > Struct Template Reference

#### Static Public Attributes

- static const bool **isValid** = false

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

### 5.309 Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::\*)(ArgT) const > Struct Template Reference

#### Public Types

- using **ArgType** = std::remove\_const\_t< std::remove\_reference\_t< ArgT > >
- using **ReturnType** = ReturnT

#### Static Public Attributes

- static const bool **isValid** = true

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

### 5.310 Catch::Detail::unique\_ptr< T > Class Template Reference

```
#include <catch_amalgamated.hpp>
```

#### Public Member Functions

- constexpr **unique\_ptr** (std::nullptr\_t=nullptr)
- constexpr **unique\_ptr** (T \*ptr)
- template<typename U , typename = std::enable\_if\_t<std::is\_base\_of<T, U>::value>> **unique\_ptr** ([unique\\_ptr](#)< U > &&from)
- template<typename U , typename = std::enable\_if\_t<std::is\_base\_of<T, U>::value>> [unique\\_ptr](#) & **operator=** ([unique\\_ptr](#)< U > &&from)
- **unique\_ptr** ([unique\\_ptr](#) const &)=delete
- [unique\\_ptr](#) & **operator=** ([unique\\_ptr](#) const &)=delete
- **unique\_ptr** ([unique\\_ptr](#) &&rhs) noexcept
- [unique\\_ptr](#) & **operator=** ([unique\\_ptr](#) &&rhs) noexcept
- T & **operator\*** ()
- T const & **operator\*** () const
- T \* **operator->** () noexcept
- T const \* **operator->** () const noexcept
- T \* **get** ()
- T const \* **get** () const
- void **reset** (T \*ptr=nullptr)
- T \* **release** ()
- **operator bool** () const

## Friends

- void **swap** ([unique\\_ptr](#) &lhs, [unique\\_ptr](#) &rhs)

### 5.310.1 Detailed Description

```
template<typename T>
class Catch::Detail::unique_ptr< T >
```

A reimplementation of `std::unique_ptr` for improved compilation performance

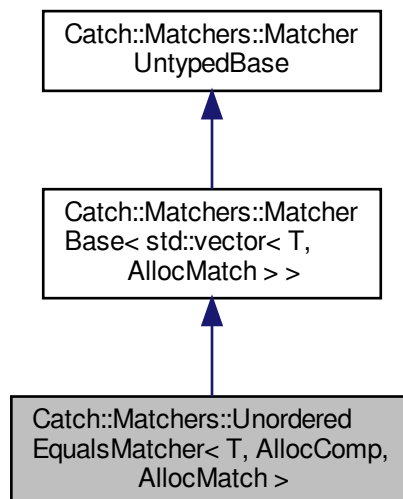
Does not support arrays nor custom deleters.

The documentation for this class was generated from the following file:

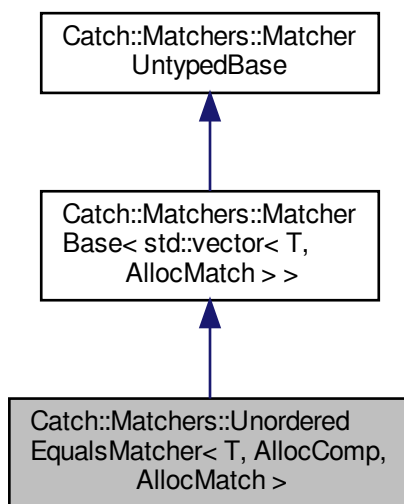
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.311 Catch::Matchers::UnorderedEqualsMatcher< T, AllocComp, AllocMatch > Class Template Reference

Inheritance diagram for Catch::Matchers::UnorderedEqualsMatcher< T, AllocComp, AllocMatch >:



Collaboration diagram for `Catch::Matchers::UnorderedEqualsMatcher< T, AllocComp, AllocMatch >`:



## Public Member Functions

- **UnorderedEqualsMatcher** (`std::vector< T, AllocComp > const &target`)
- `bool match` (`std::vector< T, AllocMatch > const &vec`) `const` override
- `std::string describe` () `const` override

## Additional Inherited Members

The documentation for this class was generated from the following file:

- `include/catch2/catch_amalgamated.hpp`

## 5.312 UserData< TSeq > Class Template Reference

Personalized data by the user.

```
#include <userdata-bones.hpp>
```

## Public Member Functions

- **UserData** ([Model](#)< TSeq > &m)
- **UserData** (std::vector< std::string > names)  
Construct a new User Data object.
- std::vector< std::string > & **get\_names** ()
- std::vector< int > & **get\_dates** ()
- std::vector< epiworld\_double > & **get\_data** ()
- void **get\_all** (std::vector< std::string > \*names=nullptr, std::vector< int > \*date=nullptr, std::vector< epiworld\_double > \*data=nullptr)
- unsigned int **nrow** () const
- unsigned int **ncol** () const
- void **write** (std::string fn)
- void **print** () const

## Append data

### Parameters

|   |                                                                                                |
|---|------------------------------------------------------------------------------------------------|
| x | A vector of length <code>ncol()</code> (if vector), otherwise a <code>epiworld_double</code> . |
| j | Index of the data point, from 0 to <code>ncol() - 1</code> .                                   |

- void **add** (std::vector< epiworld\_double > x)
- void **add** (unsigned int j, epiworld\_double x)

## Access data

### Parameters

|   |                                           |
|---|-------------------------------------------|
| i | Row (0 through <code>ndays - 1</code> .)  |
| j | Column (0 through <code>ncols()</code> ). |

### Returns

`epiworld_double&`

- `epiworld_double & operator()` (unsigned int i, unsigned int j)
- `epiworld_double & operator()` (unsigned int i, std::string name)

## Friends

- class **Model**< TSeq >
- class **DataBase**< TSeq >

## 5.312.1 Detailed Description

```
template<typename TSeq>
class UserData< TSeq >
```

Personalized data by the user.

## Template Parameters

|             |  |
|-------------|--|
| <i>TSeq</i> |  |
|-------------|--|

## 5.312.2 Constructor & Destructor Documentation

### 5.312.2.1 UserData()

```
template<typename TSeq >
UserData< TSeq >::UserData (
    std::vector< std::string > names ) [inline]
```

Construct a new User Data object.

## Parameters

|              |                                                                                   |
|--------------|-----------------------------------------------------------------------------------|
| <i>names</i> | A vector of names. The length of the vector sets the number of columns to record. |
|--------------|-----------------------------------------------------------------------------------|

The documentation for this class was generated from the following files:

- include/epiworld/database-bones.hpp
- include/epiworld/userdata-bones.hpp
- include/epiworld/userdata-meat.hpp

## 5.313 vecHasher< T > Struct Template Reference

Vector hasher.

```
#include <misc.hpp>
```

### Public Member Functions

- `std::size_t operator() (std::vector< T > const &dat) const` noexcept

### 5.313.1 Detailed Description

```
template<typename T>
struct vecHasher< T >
```

Vector hasher.

## Template Parameters

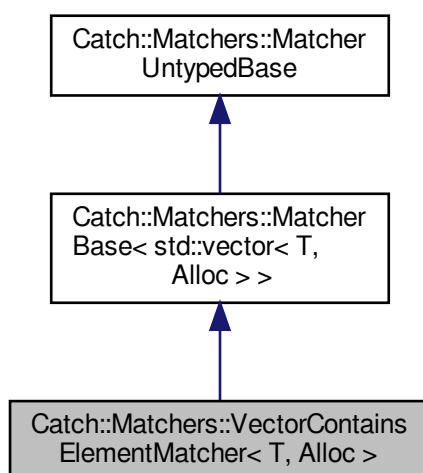
|          |  |
|----------|--|
| <i>T</i> |  |
|----------|--|

The documentation for this struct was generated from the following file:

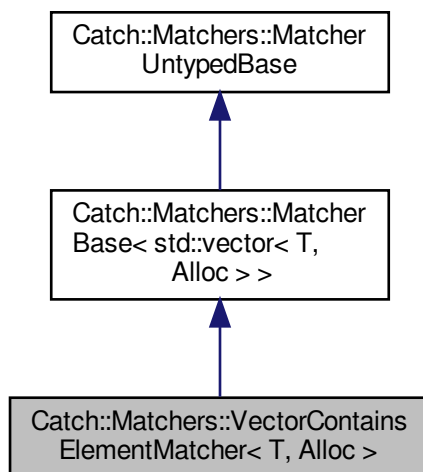
- include/epiworld/misc.hpp

## 5.314 Catch::Matchers::VectorContainsElementMatcher< T, Alloc > Class Template Reference

Inheritance diagram for Catch::Matchers::VectorContainsElementMatcher< T, Alloc >:



Collaboration diagram for `Catch::Matchers::VectorContainsElementMatcher< T, Alloc >`:



## Public Member Functions

- **VectorContainsElementMatcher** (T const &comparator)
- bool **match** (std::vector< T, Alloc > const &v) const override
- std::string **describe** () const override

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.315 Catch::Version Struct Reference

### Public Member Functions

- **Version** ([Version](#) const &)=delete
- [Version](#) & **operator=** ([Version](#) const &)=delete
- **Version** (unsigned int \_majorVersion, unsigned int \_minorVersion, unsigned int \_patchNumber, char const \*const \_branchName, unsigned int \_buildNumber)

### Public Attributes

- unsigned int const **majorVersion**
- unsigned int const **minorVersion**
- unsigned int const **patchNumber**
- char const \*const **branchName**
- unsigned int const **buildNumber**



## Friends

- `std::ostream & operator<< (std::ostream &os, Version const &version)`

The documentation for this struct was generated from the following file:

- `include/catch2/catch\_amalgamated.hpp`

## 5.316 Virus< TSeq > Class Template Reference

[Virus](#).

```
#include <virus-bones.hpp>
```

## Public Member Functions

- **Virus** (std::string name="unknown virus")
- void **mutate** ()
- void **set\_mutation** (MutFun< TSeq > fun)
- const TSeq \* **get\_sequence** ()
- void **set\_sequence** (TSeq sequence)
- [Agent](#)< TSeq > \* **get\_agent** ()
- void **set\_agent** ([Agent](#)< TSeq > \*p, epiworld\_fast\_uint idx)
- [Model](#)< TSeq > \* **get\_model** ()
- void **set\_date** (int d)
- int **get\_date** () const
- void **set\_id** (int idx)
- int **get\_id** () const
- void **set\_name** (std::string name)
- std::string **get\_name** () const
- std::vector< epiworld\_double > & **get\_data** ()

### Get and set the tool functions

#### Parameters

|     |                                        |
|-----|----------------------------------------|
| v   | <i>The virus over which to operate</i> |
| fun | <i>the function to be used</i>         |

#### Returns

*epiworld\_double*

- epiworld\_double **get\_prob\_infecting** ()
- epiworld\_double **get\_prob\_recovery** ()
- epiworld\_double **get\_prob\_death** ()
- void **post\_recovery** ()
- void **set\_post\_recovery** (PostRecoveryFun< TSeq > fun)
- void **set\_post\_immunity** (epiworld\_double prob)
- void **set\_post\_immunity** (epiworld\_double \*prob)
- void **set\_prob\_infecting\_fun** (VirusFun< TSeq > fun)

- void **set\_prob\_recovery\_fun** (VirusFun< TSeq > fun)
- void **set\_prob\_death\_fun** (VirusFun< TSeq > fun)
- void **set\_prob\_infecting** (epiworld\_double \*prob)
- void **set\_prob\_recovery** (epiworld\_double \*prob)
- void **set\_prob\_death** (epiworld\_double \*prob)
- void **set\_prob\_infecting** (epiworld\_double prob)
- void **set\_prob\_recovery** (epiworld\_double prob)
- void **set\_prob\_death** (epiworld\_double prob)

### Get and set the status and queue

After applied, viruses can change the status and affect the queue of agents. These function sets the default values, which are retrieved when adding or removing a virus does not specify a change in status or in queue.

#### Parameters

|         |                                                       |
|---------|-------------------------------------------------------|
| init    | After the virus/tool is added to the agent.           |
| end     | After the virus/tool is removed.                      |
| removed | After the agent ( <a href="#">Agent</a> ) is removed. |

- void **set\_status** (epiworld\_fast\_int init, epiworld\_fast\_int end, epiworld\_fast\_int removed=-99)
- void **set\_queue** (epiworld\_fast\_int init, epiworld\_fast\_int end, epiworld\_fast\_int removed=-99)
- void **get\_status** (epiworld\_fast\_int \*init, epiworld\_fast\_int \*end, epiworld\_fast\_int \*removed=-99)
- void **get\_queue** (epiworld\_fast\_int \*init, epiworld\_fast\_int \*end, epiworld\_fast\_int \*removed=-99)

### Friends

- class **Agent**< TSeq >
- class **Model**< TSeq >
- class **DataBase**< TSeq >
- void **default\_add\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_add\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)
- void **default\_rm\_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > \*m)

## 5.316.1 Detailed Description

```
template<typename TSeq = int>
class Virus< TSeq >
```

[Virus](#).

#### Template Parameters

|             |  |
|-------------|--|
| <i>TSeq</i> |  |
|-------------|--|

Raw transmissibility of a virus should be a function of its genetic sequence. Nonetheless, transmissibility can be reduced as a result of having one or more tools to fight the virus. Because of this, transmissibility should be a function of the agent.

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/virus-bones.hpp
- include/epiworld/virus-meat.hpp

## 5.317 Viruses< TSeq > Class Template Reference

Set of viruses (useful for building iterators)

```
#include <viruses-bones.hpp>
```

### Public Member Functions

- **Viruses** ([Agent](#)< TSeq > &p)
- std::vector< VIRUSPTR >::iterator **begin** ()
- std::vector< VIRUSPTR >::iterator **end** ()
- VIRUSPTR & **operator**() (size\_t i)
- VIRUSPTR & **operator**[] (size\_t i)
- size\_t **size** () const noexcept

### Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

#### 5.317.1 Detailed Description

```
template<typename TSeq>
class Viruses< TSeq >
```

Set of viruses (useful for building iterators)

Template Parameters

|             |  |
|-------------|--|
| <i>TSeq</i> |  |
|-------------|--|

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/viruses-bones.hpp

## 5.318 Viruses\_const< TSeq > Class Template Reference

Set of [Viruses](#) (const) (useful for iterators)

```
#include <viruses-bones.hpp>
```

## Public Member Functions

- **Viruses\_const** (const [Agent](#)< TSeq > &p)
- std::vector< VIRUSPTR >::const\_iterator **begin** ()
- std::vector< VIRUSPTR >::const\_iterator **end** ()
- const VIRUSPTR & **operator()** (size\_t i)
- const VIRUSPTR & **operator[]** (size\_t i)
- size\_t **size** () const noexcept

## Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

### 5.318.1 Detailed Description

```
template<typename TSeq>
class Viruses_const< TSeq >
```

Set of [Viruses](#) (const) (useful for iterators)

Template Parameters

|             |  |
|-------------|--|
| <i>TSeq</i> |  |
|-------------|--|

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/viruses-bones.hpp

## 5.319 Catch::WaitForKeypress Struct Reference

### Public Types

- enum **When** { **Never** , **BeforeStart** = 1 , **BeforeExit** = 2 , **BeforeStartAndExit** = BeforeStart | BeforeExit }

The documentation for this struct was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.320 Catch::WarnAbout Struct Reference

### Public Types

- enum [What](#) { **Nothing** = 0x00 , **NoAssertions** = 0x01 , **UnmatchedTestSpec** = 0x02 }

## 5.320.1 Member Enumeration Documentation

### 5.320.1.1 What

```
enum Catch::WarnAbout::What
```

#### Enumerator

|                   |                                                         |
|-------------------|---------------------------------------------------------|
| NoAssertions      | A test case or leaf section did not run any assertions. |
| UnmatchedTestSpec | A command line test spec matched no test cases.         |

The documentation for this struct was generated from the following file:

- include/catch2/catch\_amalgamated.hpp

## 5.321 Catch::WildcardPattern Class Reference

### Public Member Functions

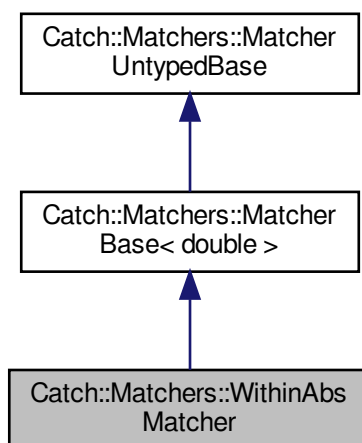
- **WildcardPattern** (std::string const &pattern, CaseSensitive caseSensitivity)
- bool **matches** (std::string const &str) const

The documentation for this class was generated from the following file:

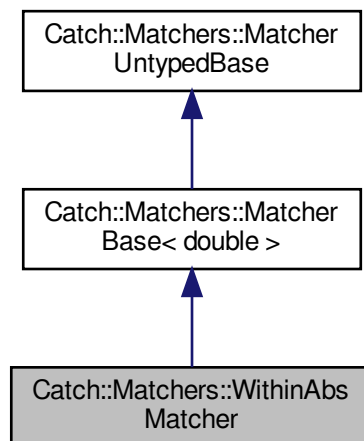
- include/catch2/catch\_amalgamated.hpp

## 5.322 Catch::Matchers::WithinAbsMatcher Class Reference

Inheritance diagram for Catch::Matchers::WithinAbsMatcher:



Collaboration diagram for Catch::Matchers::WithinAbsMatcher:



## Public Member Functions

- **WithinAbsMatcher** (double target, double margin)
- bool **match** (double const &matchee) const override
- std::string **describe** () const override

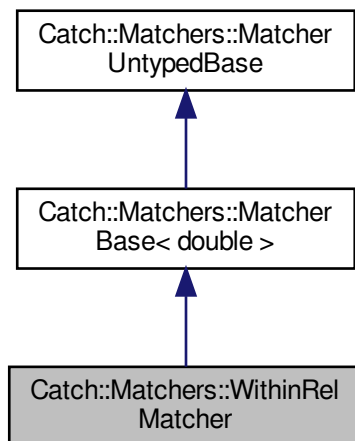
## Additional Inherited Members

The documentation for this class was generated from the following file:

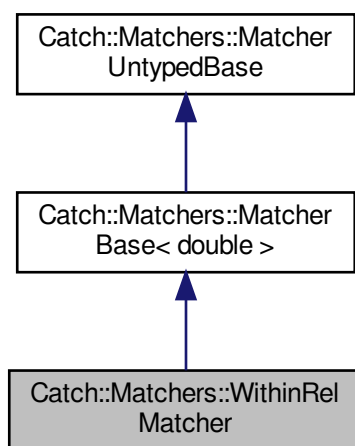
- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.323 Catch::Matchers::WithinRelMatcher Class Reference

Inheritance diagram for Catch::Matchers::WithinRelMatcher:



Collaboration diagram for Catch::Matchers::WithinRelMatcher:



### Public Member Functions

- **WithinRelMatcher** (double target, double epsilon)
- bool **match** (double const &matchee) const override
- std::string **describe** () const override

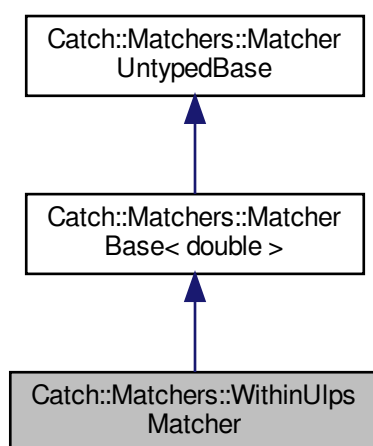
## Additional Inherited Members

The documentation for this class was generated from the following file:

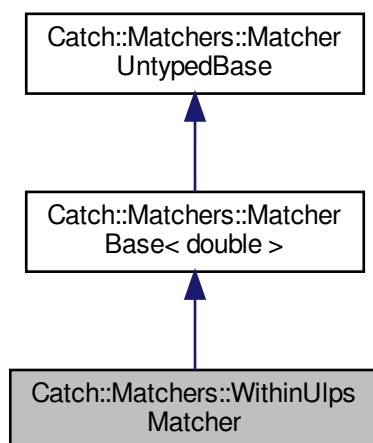
- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.324 Catch::Matchers::WithinUlpMatcher Class Reference

Inheritance diagram for Catch::Matchers::WithinUlpMatcher:



Collaboration diagram for Catch::Matchers::WithinUlpMatcher:





## Public Member Functions

- **WithinUlpMatcher** (double target, uint64\_t ulps, Detail::FloatingPointKind baseType)
- bool **match** (double const &matchee) const override
- std::string **describe** () const override

## Additional Inherited Members

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)

## 5.325 Catch::XmlEncode Class Reference

```
#include <catch_amalgamated.hpp>
```

## Public Types

- enum **ForWhat** { **ForTextNodes** , **ForAttributes** }

## Public Member Functions

- **XmlEncode** ([StringRef](#) str, ForWhat forWhat=ForTextNodes)
- void **encodeTo** (std::ostream &os) const

## Friends

- std::ostream & **operator**<< (std::ostream &os, [XmlEncode](#) const &xmlEncode)

### 5.325.1 Detailed Description

Helper for XML-encoding text (escaping angle brackets, quotes, etc)

Note: doesn't take ownership of passed strings, and thus the encoded string must outlive the encoding instance.

The documentation for this class was generated from the following file:

- include/catch2/[catch\\_amalgamated.hpp](#)



- void [sectionEnded](#) ([SectionStats](#) const &sectionStats) override  
*Called after a `SECTION` has finished running.*
- void [testCaseEnded](#) ([TestCaseStats](#) const &testCaseStats) override  
*Called once for each `TEST_CASE`, no matter how many times it is entered.*
- void [testRunEnded](#) ([TestRunStats](#) const &testRunStats) override
- void [benchmarkPreparing](#) ([StringRef](#) name) override  
*Called when user-code is being probed before the actual benchmark runs.*
- void [benchmarkStarting](#) ([BenchmarkInfo](#) const &) override  
*Called after probe but before the user-code is being benchmarked.*
- void [benchmarkEnded](#) ([BenchmarkStats](#)<> const &) override  
*Called with the benchmark results if benchmark successfully finishes.*
- void [benchmarkFailed](#) ([StringRef](#) error) override  
*Called if running the benchmarks fails for any reason.*
- void [listReporters](#) (std::vector< [ReporterDescription](#) > const &descriptions) override
- void [listListeners](#) (std::vector< [ListenerDescription](#) > const &descriptions) override
- void [listTests](#) (std::vector< [TestCaseHandle](#) > const &tests) override
- void [listTags](#) (std::vector< [TagInfo](#) > const &tags) override

## Static Public Member Functions

- static std::string [getDescription](#) ()

## Additional Inherited Members

### 5.326.1 Member Function Documentation

#### 5.326.1.1 listListeners()

```
void Catch::XmlReporter::listListeners (
    std::vector< ListenerDescription > const & descriptions ) [override], [virtual]
```

Provides a simple default listing of listeners

Looks similarly to listing of reporters, but with listener type instead of reporter name.

Reimplemented from [Catch::ReporterBase](#).

#### 5.326.1.2 listReporters()

```
void Catch::XmlReporter::listReporters (
    std::vector< ReporterDescription > const & descriptions ) [override], [virtual]
```

Provides a simple default listing of reporters.

Should look roughly like the reporter listing in v2 and earlier versions of Catch2.

Reimplemented from [Catch::ReporterBase](#).

### 5.326.1.3 listTags()

```
void Catch::XmlReporter::listTags (
    std::vector< TagInfo > const & tags ) [override], [virtual]
```

Provides a simple default listing of tags.

Should look roughly like the tag listing in v2 and earlier versions of Catch2.

Reimplemented from [Catch::ReporterBase](#).

### 5.326.1.4 listTests()

```
void Catch::XmlReporter::listTests (
    std::vector< TestCaseHandle > const & tests ) [override], [virtual]
```

Provides a simple default listing of tests.

Should look roughly like the test listing in v2 and earlier versions of Catch2. Especially supports low-verbosity listing that mimics the old `--list-test-names-only` output.

Reimplemented from [Catch::ReporterBase](#).

### 5.326.1.5 testRunEnded()

```
void Catch::XmlReporter::testRunEnded (
    TestRunStats const & testRunStats ) [override], [virtual]
```

Called once after all tests in a testing run are finished

Not called if tests weren't run (e.g. only listings happened)

Reimplemented from [Catch::StreamingReporterBase](#).

### 5.326.1.6 testRunStarting()

```
void Catch::XmlReporter::testRunStarting (
    TestRunInfo const & testRunInfo ) [override], [virtual]
```

Called once in a testing run before tests are started

Not called if tests won't be run (e.g. only listing will happen)

Reimplemented from [Catch::StreamingReporterBase](#).

The documentation for this class was generated from the following file:

- [include/catch2/catch\\_amalgamated.hpp](#)

## 5.327 Catch::XmlWriter Class Reference

### Classes

- class [ScopedElement](#)

### Public Member Functions

- **XmlWriter** (std::ostream &os)
- **XmlWriter** ([XmlWriter](#) const &)=delete
- **XmlWriter** & **operator=** ([XmlWriter](#) const &)=delete
- **XmlWriter** & **startElement** (std::string const &name, XmlFormatting fmt=XmlFormatting::Newline|XmlFormatting::Indent)
- [ScopedElement](#) **scopedElement** (std::string const &name, XmlFormatting fmt=XmlFormatting::Newline|XmlFormatting::Indent)
- **XmlWriter** & **endElement** (XmlFormatting fmt=XmlFormatting::Newline|XmlFormatting::Indent)
- **XmlWriter** & **writeAttribute** ([StringRef](#) name, [StringRef](#) attribute)  
*The attribute content is XML-encoded.*
- **XmlWriter** & **writeAttribute** ([StringRef](#) name, bool attribute)  
*Writes the attribute as "true/false".*
- **XmlWriter** & **writeAttribute** ([StringRef](#) name, char const \*attribute)  
*The attribute content is XML-encoded.*
- template<typename T, typename = typename std::enable\_if\_t< !std::is\_convertible<T, [StringRef](#)>::value>>  
**XmlWriter** & **writeAttribute** ([StringRef](#) name, T const &attribute)
- **XmlWriter** & **writeText** ([StringRef](#) text, XmlFormatting fmt=XmlFormatting::Newline|XmlFormatting::Indent)  
*Writes escaped text in a element.*
- **XmlWriter** & **writeComment** ([StringRef](#) text, XmlFormatting fmt=XmlFormatting::Newline|XmlFormatting::Indent)  
*Writes XML comment as "<!-- text -->".*
- void **writeStylesheetRef** ([StringRef](#) url)
- void **ensureTagClosed** ()

### 5.327.1 Member Function Documentation

#### 5.327.1.1 writeAttribute()

```
template<typename T, typename = typename std::enable_if_t< !std::is_convertible<T, StringRef>::value>>
XmlWriter& Catch::XmlWriter::writeAttribute (
    StringRef name,
    T const & attribute ) [inline]
```

The attribute value must provide op<<(ostream&, T). The resulting serialization is XML-encoded

The documentation for this class was generated from the following file:

- include/catch2/catch\_amalgamated.hpp



## Chapter 6

# File Documentation

### 6.1 include/catch2/catch\_amalgamated.hpp File Reference

```
#include <cstdint>
#include <string>
#include <iosfwd>
#include <cassert>
#include <chrono>
#include <vector>
#include <type_traits>
#include <cstdint>
#include <map>
#include <ratio>
#include <algorithm>
#include <iterator>
#include <numeric>
#include <tuple>
#include <cmath>
#include <functional>
#include <string.h>
#include <ostream>
#include <ctime>
#include <memory>
#include <sstream>
#include <exception>
#include <random>
#include <initializer_list>
#include <utility>
#include <limits>
#include <set>
#include <cstdio>
#include <array>
#include <cstring>
```

Include dependency graph for catch\_amalgamated.hpp:



## Classes

- class [Catch::Detail::NonCopyable](#)  
*Deriving classes become noncopyable and nonmovable.*
- class [Catch::StringRef](#)
- struct [Catch::WarnAbout](#)
- struct [Catch::WaitForKeypress](#)
- class [Catch::IConfig](#)
- class [Catch::IContext](#)
- class [Catch::IMutableContext](#)
- struct [Catch::SourceLineInfo](#)
- struct [Catch::Counts](#)
- struct [Catch::Totals](#)
- struct [Catch::SectionInfo](#)
- struct [Catch::SectionEndInfo](#)
- struct [Catch::ResultWas](#)
- struct [Catch::ResultDisposition](#)
- struct [Catch::AssertionInfo](#)
- class [Catch::LazyExpression](#)
- struct [Catch::AssertionResultData](#)
- class [Catch::AssertionResult](#)
- class [Catch::IResultCapture](#)
- struct [Catch::MessageInfo](#)
- class [Catch::Detail::unique\\_ptr< T >](#)
- struct [Catch::Benchmark::Estimate< Duration >](#)
- struct [Catch::Benchmark::OutlierClassification](#)
- struct [Catch::ReporterConfig](#)
- struct [Catch::TestRunInfo](#)
- struct [Catch::AssertionStats](#)
- struct [Catch::SectionStats](#)
- struct [Catch::TestCaseStats](#)
- struct [Catch::TestRunStats](#)
- struct [Catch::BenchmarkInfo](#)
- struct [Catch::BenchmarkStats< Duration >](#)
- struct [Catch::ReporterPreferences](#)
- class [Catch::IEventListener](#)
- struct [Catch::Benchmark::now< Clock >](#)
- struct [Catch::TestFailureException](#)  
*Used to signal that an assertion macro failed.*
- struct [Catch::always\\_false< T >](#)
- struct [Catch::true\\_given< typename >](#)
- struct [Catch::is\\_callable\\_tester](#)
- struct [Catch::is\\_callable< Fun\(Args...\)>](#)
- class [Catch::IRegistryHub](#)
- class [Catch::IMutableRegistryHub](#)
- struct [Catch::Benchmark::Detail::CompleteType< T >](#)
- struct [Catch::Benchmark::Detail::CompleteType< void >](#)
- struct [Catch::Benchmark::Detail::CompleteType< void >::type](#)
- struct [Catch::Benchmark::Detail::CompleteInvoker< Result >](#)
- struct [Catch::Benchmark::Detail::CompleteInvoker< void >](#)
- struct [Catch::Benchmark::Detail::ChronometerConcept](#)
- struct [Catch::Benchmark::Detail::ChronometerModel< Clock >](#)
- struct [Catch::Benchmark::Chronometer](#)
- struct [Catch::Benchmark::EnvironmentEstimate< Duration >](#)



- struct [Catch::Benchmark::Environment< Clock >](#)
- struct [Catch::Benchmark::Detail::is\\_related< T, U >](#)
- struct [Catch::Benchmark::Detail::BenchmarkFunction](#)
- struct [Catch::Benchmark::Detail::repeater< Fun >](#)
- struct [Catch::Benchmark::Timing< Duration, Result >](#)
- struct [Catch::Benchmark::ExecutionPlan< Duration >](#)
- struct [Catch::Benchmark::Detail::bootstrap\\_analysis](#)
- struct [Catch::Benchmark::SampleAnalysis< Duration >](#)
- struct [Catch::Benchmark::Benchmark](#)
- struct [Catch::Benchmark::Detail::ObjectStorage< T, Destruct >](#)
- class [Catch::ReusableStringStream](#)
- struct [Catch::Detail::make\\_void<... >](#)
- struct [Catch::Detail::EnumInfo](#)
- class [Catch::MutableEnumValuesRegistry](#)
- struct [Catch\\_global\\_namespace\\_dummy](#)
- class [Catch::Detail::IsStreamInsertable< T >](#)
- struct [Catch::StringMaker< T, typename >](#)
- struct [Catch::StringMaker< std::string >](#)
- struct [Catch::StringMaker< char const \\* >](#)
- struct [Catch::StringMaker< char \\* >](#)
- struct [Catch::StringMaker< std::wstring >](#)
- struct [Catch::StringMaker< wchar\\_t const \\* >](#)
- struct [Catch::StringMaker< wchar\\_t \\* >](#)
- struct [Catch::StringMaker< char\[SZ\]>](#)
- struct [Catch::StringMaker< signed char\[SZ\]>](#)
- struct [Catch::StringMaker< unsigned char\[SZ\]>](#)
- struct [Catch::StringMaker< int >](#)
- struct [Catch::StringMaker< long >](#)
- struct [Catch::StringMaker< long long >](#)
- struct [Catch::StringMaker< unsigned int >](#)
- struct [Catch::StringMaker< unsigned long >](#)
- struct [Catch::StringMaker< unsigned long long >](#)
- struct [Catch::StringMaker< bool >](#)
- struct [Catch::StringMaker< char >](#)
- struct [Catch::StringMaker< signed char >](#)
- struct [Catch::StringMaker< unsigned char >](#)
- struct [Catch::StringMaker< std::nullptr\\_t >](#)
- struct [Catch::StringMaker< float >](#)
- struct [Catch::StringMaker< double >](#)
- struct [Catch::StringMaker< T \\* >](#)
- struct [Catch::StringMaker< R C::\\* >](#)
- struct [Catch::Detail::is\\_range\\_impl< T, typename >](#)
- struct [Catch::Detail::is\\_range\\_impl< T, void\\_t< decltype\(begin\(std::declval< T >\(\)\)\)> >](#)
- struct [Catch::is\\_range< T >](#)
- struct [Catch::StringMaker< R, std::enable\\_if\\_t< is\\_range< R >::value &&!::Catch::Detail::IsStreamInsertable< R >::value >](#)
- struct [Catch::StringMaker< T\[SZ\]>](#)
- struct [Catch::ratio\\_string< Ratio >](#)
- struct [Catch::ratio\\_string< std::atto >](#)
- struct [Catch::ratio\\_string< std::femto >](#)
- struct [Catch::ratio\\_string< std::pico >](#)
- struct [Catch::ratio\\_string< std::nano >](#)
- struct [Catch::ratio\\_string< std::micro >](#)
- struct [Catch::ratio\\_string< std::milli >](#)
- struct [Catch::StringMaker< std::chrono::duration< Value, Ratio > >](#)
- struct [Catch::StringMaker< std::chrono::duration< Value, std::ratio< 1 > > >](#)

- struct [Catch::StringMaker< std::chrono::duration< Value, std::ratio< 60 > > >](#)
- struct [Catch::StringMaker< std::chrono::duration< Value, std::ratio< 3600 > > >](#)
- struct [Catch::StringMaker< std::chrono::time\\_point< Clock, Duration > >](#)
- struct [Catch::StringMaker< std::chrono::time\\_point< std::chrono::system\\_clock, Duration > >](#)
- class [Catch::Approx](#)
- struct [Catch::StringMaker< Catch::Approx >](#)
- class [Catch::WildcardPattern](#)
- class [Catch::TestSpec](#)
- struct [Catch::TestSpec::FilterMatch](#)
- class [Catch::Optional< T >](#)
- struct [Catch::Colour](#)
- class [Catch::ColourImpl](#)
- class [Catch::ColourImpl::ColourGuard](#)
- class [Catch::ReporterSpec](#)
- struct [Catch::ProcessedReporterSpec](#)
- struct [Catch::ConfigData](#)
- class [Catch::Config](#)
- struct [Catch::StreamEndStop](#)
- struct [Catch::MessageStream](#)
- struct [Catch::MessageBuilder](#)
- class [Catch::ScopedMessage](#)
- class [Catch::Capturer](#)
- struct [Catch::Clara::accept\\_many\\_t](#)
- struct [Catch::Clara::Detail::fake\\_arg](#)
- struct [Catch::Clara::Detail::is\\_unary\\_function< F, typename >](#)
- struct [Catch::Clara::Detail::is\\_unary\\_function< F, Catch::Detail::void\\_t< decltype\(std::declval< F >\(\)\(fake\\_arg\(\)\)\) > >](#)
- struct [Catch::Clara::Detail::UnaryLambdaTraits< L >](#)
- struct [Catch::Clara::Detail::UnaryLambdaTraits< ReturnT\(ClassT::\\*\)\(Args...\) const >](#)
- struct [Catch::Clara::Detail::UnaryLambdaTraits< ReturnT\(ClassT::\\*\)\(ArgT\) const >](#)
- struct [Catch::Clara::Detail::Token](#)
- class [Catch::Clara::Detail::TokenStream](#)
- class [Catch::Clara::Detail::ResultBase](#)
- class [Catch::Clara::Detail::ResultValueBase< T >](#)
- class [Catch::Clara::Detail::ResultValueBase< void >](#)
- class [Catch::Clara::Detail::BasicResult< T >](#)
- class [Catch::Clara::Detail::ParseState](#)
- struct [Catch::Clara::Detail::HelpColumns](#)
- struct [Catch::Clara::Detail::BoundRef](#)
- struct [Catch::Clara::Detail::BoundValueRefBase](#)
- struct [Catch::Clara::Detail::BoundFlagRefBase](#)
- struct [Catch::Clara::Detail::BoundValueRef< T >](#)
- struct [Catch::Clara::Detail::BoundValueRef< std::vector< T > >](#)
- struct [Catch::Clara::Detail::BoundFlagRef](#)
- struct [Catch::Clara::Detail::LambdaInvoker< ReturnType >](#)
- struct [Catch::Clara::Detail::LambdaInvoker< void >](#)
- struct [Catch::Clara::Detail::BoundLambda< L >](#)
- struct [Catch::Clara::Detail::BoundManyLambda< L >](#)
- struct [Catch::Clara::Detail::BoundFlagLambda< L >](#)
- class [Catch::Clara::Detail::ParserBase](#)
- class [Catch::Clara::Detail::ComposableParserImpl< DerivedT >](#)
- class [Catch::Clara::Detail::ParserRefImpl< DerivedT >](#)
- class [Catch::Clara::Arg](#)
- class [Catch::Clara::Opt](#)
- class [Catch::Clara::ExeName](#)
- class [Catch::Clara::Parser](#)

- class [Catch::Clara::Args](#)
- struct [Catch::Clara::Help](#)
- class [Catch::Session](#)
- struct [Catch::TagAlias](#)
- struct [Catch::RegistrarForTagAliases](#)
- class [Catch::ITransientExpression](#)
- class [Catch::BinaryExpr< LhsT, RhsT >](#)
- class [Catch::UnaryExpr< LhsT >](#)
- class [Catch::ExprLhs< LhsT >](#)
- struct [Catch::Decomposer](#)
- struct [Catch::AssertionReaction](#)
- class [Catch::AssertionHandler](#)
- class [Catch::Timer](#)
- class [Catch::Section](#)
- class [Catch::ITestInvoker](#)
- class [Catch::ITestCaseRegistry](#)
- class [Catch::TestInvokerAsMethod< C >](#)
- struct [Catch::NameAndTags](#)
- struct [Catch::AutoReg](#)
- struct [Catch::Tag](#)
- struct [Catch::TestCaseInfo](#)
- class [Catch::TestCaseHandle](#)
- class [Catch::IExceptionTranslator](#)
- class [Catch::IExceptionTranslatorRegistry](#)
- class [Catch::ExceptionTranslatorRegistrar](#)
- struct [Catch::Version](#)
- class [Catch::GeneratorException](#)
- class [Catch::Generators::GeneratorUntypedBase](#)
- class [Catch::IGeneratorTracker](#)
- class [Catch::Generators::IGenerator< T >](#)
- class [Catch::Generators::GeneratorWrapper< T >](#)
- class [Catch::Generators::SingleValueGenerator< T >](#)
- class [Catch::Generators::FixedValuesGenerator< T >](#)
- class [Catch::Generators::Generators< T >](#)
- struct [Catch::Generators::as< T >](#)
- class [Catch::Generators::TakeGenerator< T >](#)
- class [Catch::Generators::FilterGenerator< T, Predicate >](#)
- class [Catch::Generators::RepeatGenerator< T >](#)
- class [Catch::Generators::MapGenerator< T, U, Func >](#)
- class [Catch::Generators::ChunkGenerator< T >](#)
- class [Catch::SimplePcg32](#)
- class [Catch::Generators::RandomFloatingGenerator< Float >](#)
- class [Catch::Generators::RandomIntegerGenerator< Integer >](#)
- class [Catch::Generators::RangeGenerator< T >](#)
- class [Catch::Generators::IteratorGenerator< T >](#)
- class [Catch::IReporterFactory](#)
- class [Catch::EventListenerFactory](#)
- struct [Catch::Detail::CaseInsensitiveLess](#)  
*Provides case-insensitive op< semantics when called.*
- struct [Catch::Detail::CaseInsensitiveEqualTo](#)  
*Provides case-insensitive op== semantics when called.*
- class [Catch::IReporterRegistry](#)
- class [Catch::ITagAliasRegistry](#)
- class [Catch::Detail::EnumValuesRegistry](#)

- class [Catch::ErrnoGuard](#)
- class [Catch::ExceptionTranslatorRegistry](#)
- class [Catch::FatalConditionHandler](#)
- class [Catch::FatalConditionHandlerGuard](#)
  - Simple RAI guard for (dis)engaging the FatalConditionHandler.*
- class [Catch::IStream](#)
- struct [Catch::LeakDetector](#)
- struct [Catch::ReporterDescription](#)
- struct [Catch::ListenerDescription](#)
- struct [Catch::TagInfo](#)
- class [Catch::RedirectedStream](#)
- class [Catch::RedirectedStdOut](#)
- class [Catch::RedirectedStdErr](#)
- class [Catch::RedirectedStreams](#)
- class [Catch::ReporterRegistry](#)
- struct [Catch::TestCaseTracking::NameAndLocation](#)
- class [Catch::TestCaseTracking::ITracker](#)
- class [Catch::TestCaseTracking::TrackerContext](#)
- class [Catch::TestCaseTracking::TrackerBase](#)
- class [Catch::TestCaseTracking::SectionTracker](#)
- class [Catch::RunContext](#)
- struct [Catch::ISingleton](#)
- class [Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT >](#)
- class [Catch::StartupExceptionRegistry](#)
- class [Catch::pluralise](#)
- class [Catch::TagAliasRegistry](#)
- class [Catch::TestCaseInfoHasher](#)
- class [Catch::TestRegistry](#)
- class [Catch::TestInvokerAsFunction](#)
- class [Catch::TestSpecParser](#)
- class [Catch::TextFlow::Column](#)
- class [Catch::TextFlow::Column::const\\_iterator](#)
- class [Catch::TextFlow::Columns](#)
- class [Catch::TextFlow::Columns::iterator](#)
- class [Catch::XmlEncode](#)
- class [Catch::XmlWriter](#)
- class [Catch::XmlWriter::ScopedElement](#)
- class [Catch::MatchExpr< ArgT, MatcherT >](#)
- class [Catch::Matchers::MatcherUntypedBase](#)
- class [Catch::Matchers::MatcherBase< T >](#)
- class [Catch::Matchers::Detail::MatchAllOf< ArgT >](#)
- class [Catch::Matchers::Detail::MatchAnyOf< ArgT >](#)
- class [Catch::Matchers::Detail::MatchNotOf< ArgT >](#)
- class [Catch::Matchers::MatcherGenericBase](#)
- struct [Catch::Matchers::Detail::conjunction< Cond >](#)
- struct [Catch::Matchers::Detail::conjunction< Cond, Rest... >](#)
- class [Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs >](#)
- class [Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs >](#)
- class [Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT >](#)
- class [Catch::Matchers::IsEmptyMatcher](#)
- class [Catch::Matchers::HasSizeMatcher](#)
- class [Catch::Matchers::SizeMatchesMatcher< Matcher >](#)
- class [Catch::Matchers::ContainsElementMatcher< T, Equality >](#)

*Matcher for checking that an element in range is equal to specific element.*

- class [Catch::Matchers::ContainsMatcher](#)[Matcher< Matcher >](#)  
*Meta-matcher for checking that an element in a range matches a specific matcher.*
- class [Catch::Matchers::ExceptionMessageMatcher](#)
- class [Catch::Matchers::WithinAbsMatcher](#)
- class [Catch::Matchers::WithinUlpMatcher](#)
- class [Catch::Matchers::WithinRelMatcher](#)
- class [Catch::Matchers::PredicateMatcher](#)[< T, Predicate >](#)
- class [Catch::Matchers::AllMatchMatcher](#)[< Matcher >](#)
- class [Catch::Matchers::NoneMatchMatcher](#)[< Matcher >](#)
- class [Catch::Matchers::AnyMatchMatcher](#)[< Matcher >](#)
- struct [Catch::Matchers::CasedString](#)
- class [Catch::Matchers::StringMatcherBase](#)
- class [Catch::Matchers::StringEqualsMatcher](#)
- class [Catch::Matchers::StringContainsMatcher](#)
- class [Catch::Matchers::StartsWithMatcher](#)
- class [Catch::Matchers::EndsWithMatcher](#)
- class [Catch::Matchers::RegexMatcher](#)
- class [Catch::Matchers::VectorContainsElementMatcher](#)[< T, Alloc >](#)
- class [Catch::Matchers::ContainsMatcher](#)[< T, AllocComp, AllocMatch >](#)
- class [Catch::Matchers::EqualsMatcher](#)[< T, AllocComp, AllocMatch >](#)
- class [Catch::Matchers::ApproxMatcher](#)[< T, AllocComp, AllocMatch >](#)
- class [Catch::Matchers::UnorderedEqualsMatcher](#)[< T, AllocComp, AllocMatch >](#)
- class [Catch::ReporterBase](#)
- class [Catch::StreamingReporterBase](#)
- class [Catch::AutomakeReporter](#)
- class [Catch::CompactReporter](#)
- class [Catch::ConsoleReporter](#)
- class [Catch::Detail::AssertionOrBenchmarkResult](#)  
*Represents either an assertion or a benchmark result to be handled by cumulative reporter later.*
- class [Catch::CumulativeReporterBase](#)
- struct [Catch::CumulativeReporterBase::Node](#)[< T, ChildNodeT >](#)
- struct [Catch::CumulativeReporterBase::SectionNode](#)
- class [Catch::EventListenerBase](#)
- struct [Catch::lineOfChars](#)
- class [Catch::JUnitReporter](#)
- class [Catch::MultiReporter](#)
- struct [Catch::Detail::has\\_description](#)[< T, typename >](#)
- struct [Catch::Detail::has\\_description](#)[< T, void\\_t< decltype\(T::getDescription\(\)\)> >](#)
- class [Catch::ReporterFactory](#)[< T >](#)
- class [Catch::ReporterRegistrar](#)[< T >](#)
- class [Catch::ListenerRegistrar](#)[< T >](#)
- class [Catch::SonarQubeReporter](#)
- class [Catch::TAPReporter](#)
- class [Catch::TeamCityReporter](#)
- class [Catch::XmlReporter](#)

## Macros

- `#define CATCH_ALL_HPP_INCLUDED`
- `#define CATCH_BENCHMARK_ALL_HPP_INCLUDED`
- `#define CATCH_BENCHMARK_HPP_INCLUDED`
- `#define CATCH_INTERFACES_CONFIG_HPP_INCLUDED`
- `#define CATCH_NONCOPYABLE_HPP_INCLUDED`
- `#define CATCH_STRINGREF_HPP_INCLUDED`
- `#define CATCH_COMPILER_CAPABILITIES_HPP_INCLUDED`
- `#define CATCH_PLATFORM_HPP_INCLUDED`
- `#define CATCH_INTERNAL_CONFIG_POSIX_SIGNALS`
- `#define CATCH_INTERNAL_CONFIG_GLOBAL_NEXTAFTER`
- `#define CATCH_CONFIG_POSIX_SIGNALS`
- `#define CATCH_CONFIG_CPP11_TO_STRING`
- `#define CATCH_CONFIG_DISABLE_EXCEPTIONS`
- `#define CATCH_CONFIG_GLOBAL_NEXTAFTER`
- `#define CATCH_INTERNAL_START_WARNINGS_SUPPRESSION`
- `#define CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION`
- `#define CATCH_INTERNAL_SUPPRESS_PARENTHESES_WARNINGS`
- `#define CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS`
- `#define CATCH_INTERNAL_SUPPRESS_UNUSED_VARIABLE_WARNINGS`
- `#define CATCH_INTERNAL_SUPPRESS_ZERO_VARIADIC_WARNINGS`
- `#define CATCH_INTERNAL_IGNORE_BUT_WARN(...)`
- `#define CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS`
- `#define CATCH_TRY if ((true))`
- `#define CATCH_CATCH_ALL if ((false))`
- `#define CATCH_CATCH_ANON(type) if ((false))`
- `#define CATCH_CONTEXT_HPP_INCLUDED`
- `#define CATCH_INTERFACES_REPORTER_HPP_INCLUDED`
- `#define CATCH_SECTION_INFO_HPP_INCLUDED`
- `#define CATCH_MOVE_AND_FORWARD_HPP_INCLUDED`
- `#define CATCH_MOVE(...) static_cast<std::remove_reference_t<decltype(__VA_ARGS__)>&&>(__VA_ARGS__)`  
*Replacement for std::move with better compile time performance.*
- `#define CATCH_FORWARD(...) static_cast<decltype(__VA_ARGS__)&&>(__VA_ARGS__)`  
*Replacement for std::forward with better compile time performance.*
- `#define CATCH_SOURCE_LINE_INFO_HPP_INCLUDED`
- `#define CATCH_INTERNAL_LINEINFO ::Catch::SourceLineInfo( __FILE__, static_cast<std::size_t> ( __`  
`_LINE__ ) )`
- `#define CATCH_TOTALS_HPP_INCLUDED`
- `#define CATCH_ASSERTION_RESULT_HPP_INCLUDED`
- `#define CATCH_ASSERTION_INFO_HPP_INCLUDED`
- `#define CATCH_RESULT_TYPE_HPP_INCLUDED`
- `#define CATCH_LAZY_EXPR_HPP_INCLUDED`
- `#define CATCH_MESSAGE_INFO_HPP_INCLUDED`
- `#define CATCH_INTERFACES_CAPTURE_HPP_INCLUDED`
- `#define CATCH_UNIQUE_PTR_HPP_INCLUDED`
- `#define CATCH_ESTIMATE_HPP_INCLUDED`
- `#define CATCH_OUTLIER_CLASSIFICATION_HPP_INCLUDED`
- `#define CATCH_UNIQUE_NAME_HPP_INCLUDED`
- `#define CATCH_CONFIG_COUNTER_HPP_INCLUDED`
- `#define CATCH_INTERNAL_CONFIG_COUNTER`
- `#define CATCH_CONFIG_COUNTER`
- `#define INTERNAL_CATCH_UNIQUE_NAME_LINE2(name, line) name##line`

- #define **INTERNAL\_CATCH\_UNIQUE\_NAME\_LINE**(name, line) INTERNAL\_CATCH\_UNIQUE\_NAME\_  
LINE2( name, line )
- #define **INTERNAL\_CATCH\_UNIQUE\_NAME**(name) INTERNAL\_CATCH\_UNIQUE\_NAME\_LINE( name,  
\_\_COUNTER\_\_ )
- #define **CATCH\_CHRONOMETER\_HPP\_INCLUDED**
- #define **CATCH\_CLOCK\_HPP\_INCLUDED**
- #define **CATCH\_OPTIMIZER\_HPP\_INCLUDED**
- #define **CATCH\_COMPLETE\_INVOKE\_HPP\_INCLUDED**
- #define **CATCH\_TEST\_FAILURE\_EXCEPTION\_HPP\_INCLUDED**
- #define **CATCH\_META\_HPP\_INCLUDED**
- #define **CATCH\_INTERFACES\_REGISTRY\_HUB\_HPP\_INCLUDED**
- #define **CATCH\_ENVIRONMENT\_HPP\_INCLUDED**
- #define **CATCH\_EXECUTION\_PLAN\_HPP\_INCLUDED**
- #define **CATCH\_BENCHMARK\_FUNCTION\_HPP\_INCLUDED**
- #define **CATCH\_REPEAT\_HPP\_INCLUDED**
- #define **CATCH\_RUN\_FOR\_AT\_LEAST\_HPP\_INCLUDED**
- #define **CATCH\_MEASURE\_HPP\_INCLUDED**
- #define **CATCH\_TIMING\_HPP\_INCLUDED**
- #define **CATCH\_ESTIMATE\_CLOCK\_HPP\_INCLUDED**
- #define **CATCH\_STATS\_HPP\_INCLUDED**
- #define **CATCH\_ANALYSE\_HPP\_INCLUDED**
- #define **CATCH\_SAMPLE\_ANALYSIS\_HPP\_INCLUDED**
- #define **INTERNAL\_CATCH\_GET\_1\_ARG**(arg1, arg2, ...) arg1
- #define **INTERNAL\_CATCH\_GET\_2\_ARG**(arg1, arg2, ...) arg2
- #define **INTERNAL\_CATCH\_BENCHMARK**(BenchmarkName, name, benchmarkIndex)
- #define **INTERNAL\_CATCH\_BENCHMARK\_ADVANCED**(BenchmarkName, name)
- #define **BENCHMARK**(...) INTERNAL\_CATCH\_BENCHMARK(INTERNAL\_CATCH\_UNIQUE\_  
NAME(CATCH2\_INTERNAL\_BENCHMARK\_), INTERNAL\_CATCH\_GET\_1\_ARG(\_\_VA\_ARGS\_\_),  
INTERNAL\_CATCH\_GET\_2\_ARG(\_\_VA\_ARGS\_\_),)
- #define **BENCHMARK\_ADVANCED**(name) INTERNAL\_CATCH\_BENCHMARK\_ADVANCED(INTERNAL\_  
CATCH\_UNIQUE\_NAME(CATCH2\_INTERNAL\_BENCHMARK\_), name)
- #define **CATCH\_CONSTRUCTOR\_HPP\_INCLUDED**
- #define **CATCH\_APPROX\_HPP\_INCLUDED**
- #define **CATCH\_TOSTRING\_HPP\_INCLUDED**
- #define **CATCH\_CONFIG\_WCHAR\_HPP\_INCLUDED**
- #define **CATCH\_CONFIG\_WCHAR**
- #define **CATCH\_REUSABLE\_STRING\_STREAM\_HPP\_INCLUDED**
- #define **CATCH\_VOID\_TYPE\_HPP\_INCLUDED**
- #define **CATCH\_INTERFACES\_ENUM\_VALUES\_REGISTRY\_HPP\_INCLUDED**
- #define **INTERNAL\_CATCH\_REGISTER\_ENUM**(enumName, ...)
- #define **CATCH\_REGISTER\_ENUM**(enumName, ...) INTERNAL\_CATCH\_REGISTER\_ENUM( enumName,  
\_\_VA\_ARGS\_\_ )
- #define **CATCH\_CONFIG\_HPP\_INCLUDED**
- #define **CATCH\_TEST\_SPEC\_HPP\_INCLUDED**
- #define **CATCH\_WILDCARD\_PATTERN\_HPP\_INCLUDED**
- #define **CATCH\_CASE\_SENSITIVE\_HPP\_INCLUDED**
- #define **CATCH\_OPTIONAL\_HPP\_INCLUDED**
- #define **CATCH\_RANDOM\_SEED\_GENERATION\_HPP\_INCLUDED**
- #define **CATCH\_REPORTER\_SPEC\_PARSER\_HPP\_INCLUDED**
- #define **CATCH\_CONSOLE\_COLOUR\_HPP\_INCLUDED**
- #define **CATCH\_MESSAGE\_HPP\_INCLUDED**
- #define **CATCH\_STREAM\_END\_STOP\_HPP\_INCLUDED**
- #define **INTERNAL\_CATCH\_MSG**(macroName, messageType, resultDisposition, ...)
- #define **INTERNAL\_CATCH\_CAPTURE**(varName, macroName, ...)

- `#define INTERNAL_CATCH_INFO(macroName, log) Catch::ScopedMessage INTERNAL_CATCH_↵  
_UNIQUE_NAME( scopedMessage )( Catch::MessageBuilder( macroName##_catch_sr, CATCH_↵  
INTERNAL_LINEINFO, Catch::ResultWas::Info ) << log )`
- `#define INTERNAL_CATCH_UNSCOPED_INFO(macroName, log) Catch::getResultCapture().emplace↵  
UnscopedMessage( Catch::MessageBuilder( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,  
Catch::ResultWas::Info ) << log )`
- `#define INFO(msg) INTERNAL_CATCH_INFO( "INFO", msg )`
- `#define UNSCOPED_INFO(msg) INTERNAL_CATCH_UNSCOPED_INFO( "UNSCOPED_INFO", msg )`
- `#define WARN(msg) INTERNAL_CATCH_MSG( "WARN", Catch::ResultWas::Warning, Catch::Result↵  
Disposition::ContinueOnFailure, msg )`
- `#define CAPTURE(...) INTERNAL_CATCH_CAPTURE( INTERNAL_CATCH_UNIQUE_NAME(capturer),  
"CAPTURE", __VA_ARGS__ )`
- `#define CATCH_SESSION_HPP_INCLUDED`
- `#define CATCH_COMMANDLINE_HPP_INCLUDED`
- `#define CATCH_CLARA_HPP_INCLUDED`
- `#define CATCH_TAG_ALIAS_HPP_INCLUDED`
- `#define CATCH_TAG_ALIAS_AUTOREGISTRAR_HPP_INCLUDED`
- `#define CATCH_REGISTER_TAG_ALIAS(alias, spec)`
- `#define CATCH_TEMPLATE_TEST_MACROS_HPP_INCLUDED`
- `#define CATCH_TEST_MACROS_HPP_INCLUDED`
- `#define CATCH_TEST_MACRO_IMPL_HPP_INCLUDED`
- `#define CATCH_ASSERTION_HANDLER_HPP_INCLUDED`
- `#define CATCH_DECOMPOSER_HPP_INCLUDED`
- `#define CATCH_INTERNAL_DEFINE_EXPRESSION_OPERATOR(op)`
- `#define CATCH_INTERNAL_STRINGIFY(...) #__VA_ARGS__`
- `#define INTERNAL_CATCH_TRY`
- `#define INTERNAL_CATCH_CATCH(capturer)`
- `#define INTERNAL_CATCH_REACT(handler) handler.complete();`
- `#define INTERNAL_CATCH_TEST(macroName, resultDisposition, ...)`
- `#define INTERNAL_CATCH_IF(macroName, resultDisposition, ...)`
- `#define INTERNAL_CATCH_ELSE(macroName, resultDisposition, ...)`
- `#define INTERNAL_CATCH_NO_THROW(macroName, resultDisposition, ...)`
- `#define INTERNAL_CATCH_THROWS(macroName, resultDisposition, ...)`
- `#define INTERNAL_CATCH_THROWS_AS(macroName, exceptionType, resultDisposition, expr)`
- `#define INTERNAL_CATCH_THROWS_STR_MATCHES(macroName, resultDisposition, matcher, ...)`
- `#define CATCH_SECTION_HPP_INCLUDED`
- `#define CATCH_TIMER_HPP_INCLUDED`
- `#define INTERNAL_CATCH_SECTION(...)`
- `#define INTERNAL_CATCH_DYNAMIC_SECTION(...)`
- `#define CATCH_TEST_REGISTRY_HPP_INCLUDED`
- `#define CATCH_INTERFACES_TESTCASE_HPP_INCLUDED`
- `#define CATCH_PREPROCESSOR_REMOVE_PARENS_HPP_INCLUDED`
- `#define INTERNAL_CATCH_EXPAND1(param) INTERNAL_CATCH_EXPAND2( param )`
- `#define INTERNAL_CATCH_EXPAND2(...) INTERNAL_CATCH_NO##_VA_ARGS__`
- `#define INTERNAL_CATCH_DEF(...) INTERNAL_CATCH_DEF __VA_ARGS__`
- `#define INTERNAL_CATCH_NOINTERNAL_CATCH_DEF`
- `#define INTERNAL_CATCH_REMOVE_PARENS(...) INTERNAL_CATCH_EXPAND1( INTERNAL_↵  
CATCH_DEF __VA_ARGS__ )`
- `#define INTERNAL_CATCH_TESTCASE2( testName, ...)`
- `#define INTERNAL_CATCH_TESTCASE(...) INTERNAL_CATCH_TESTCASE2( INTERNAL_CATCH_↵  
UNIQUE_NAME( CATCH2_INTERNAL_TEST_ ), __VA_ARGS__ )`
- `#define INTERNAL_CATCH_METHOD_AS_TEST_CASE( QualifiedMethod, ...)`
- `#define INTERNAL_CATCH_TEST_CASE_METHOD2( testName, className, ...)`
- `#define INTERNAL_CATCH_TEST_CASE_METHOD( className, ... ) INTERNAL_CATCH_TEST_↵  
CASE_METHOD2( INTERNAL_CATCH_UNIQUE_NAME( CATCH2_INTERNAL_TEST_ ), className, ↵  
__VA_ARGS__ )`



- `#define INTERNAL_CATCH_REGISTER_TESTCASE(Function, ...)`
- `#define REQUIRE(...) INTERNAL_CATCH_TEST( "REQUIRE", Catch::ResultDisposition::Normal, __VA_ARGS__ )`
- `#define REQUIRE_FALSE(...) INTERNAL_CATCH_TEST( "REQUIRE_FALSE", Catch::ResultDisposition::Normal | Catch::ResultDisposition::FalseTest, __VA_ARGS__ )`
- `#define REQUIRE_THROWS(...) INTERNAL_CATCH_THROWS( "REQUIRE_THROWS", Catch::ResultDisposition::Normal, __VA_ARGS__ )`
- `#define REQUIRE_THROWS_AS(expr, exceptionType) INTERNAL_CATCH_THROWS_AS( "REQUIRE_THROWS_AS", exceptionType, Catch::ResultDisposition::Normal, expr )`
- `#define REQUIRE_NOTHROW(...) INTERNAL_CATCH_NO_THROW( "REQUIRE_NOTHROW", Catch::ResultDisposition::Normal, __VA_ARGS__ )`
- `#define CHECK(...) INTERNAL_CATCH_TEST( "CHECK", Catch::ResultDisposition::ContinueOnFailure, __VA_ARGS__ )`
- `#define CHECK_FALSE(...) INTERNAL_CATCH_TEST( "CHECK_FALSE", Catch::ResultDisposition::ContinueOnFailure | Catch::ResultDisposition::FalseTest, __VA_ARGS__ )`
- `#define CHECKED_IF(...) INTERNAL_CATCH_IF( "CHECKED_IF", Catch::ResultDisposition::ContinueOnFailure | Catch::ResultDisposition::SuppressFail, __VA_ARGS__ )`
- `#define CHECKED_ELSE(...) INTERNAL_CATCH_ELSE( "CHECKED_ELSE", Catch::ResultDisposition::ContinueOnFailure | Catch::ResultDisposition::SuppressFail, __VA_ARGS__ )`
- `#define CHECK_NOFAIL(...) INTERNAL_CATCH_TEST( "CHECK_NOFAIL", Catch::ResultDisposition::ContinueOnFailure | Catch::ResultDisposition::SuppressFail, __VA_ARGS__ )`
- `#define CHECK_THROWS(...) INTERNAL_CATCH_THROWS( "CHECK_THROWS", Catch::ResultDisposition::ContinueOnFailure, __VA_ARGS__ )`
- `#define CHECK_THROWS_AS(expr, exceptionType) INTERNAL_CATCH_THROWS_AS( "CHECK_THROWS_AS", exceptionType, Catch::ResultDisposition::ContinueOnFailure, expr )`
- `#define CHECK_NOTHROW(...) INTERNAL_CATCH_NO_THROW( "CHECK_NOTHROW", Catch::ResultDisposition::ContinueOnFailure, __VA_ARGS__ )`
- `#define TEST_CASE(...) INTERNAL_CATCH_TESTCASE( __VA_ARGS__ )`
- `#define TEST_CASE_METHOD(className, ...) INTERNAL_CATCH_TEST_CASE_METHOD( className, __VA_ARGS__ )`
- `#define METHOD_AS_TEST_CASE(method, ...) INTERNAL_CATCH_METHOD_AS_TEST_CASE( method, __VA_ARGS__ )`
- `#define REGISTER_TEST_CASE(Function, ...) INTERNAL_CATCH_REGISTER_TESTCASE( Function, __VA_ARGS__ )`
- `#define SECTION(...) INTERNAL_CATCH_SECTION( __VA_ARGS__ )`
- `#define DYNAMIC_SECTION(...) INTERNAL_CATCH_DYNAMIC_SECTION( __VA_ARGS__ )`
- `#define FAIL(...) INTERNAL_CATCH_MSG( "FAIL", Catch::ResultWas::ExplicitFailure, Catch::ResultDisposition::Normal, __VA_ARGS__ )`
- `#define FAIL_CHECK(...) INTERNAL_CATCH_MSG( "FAIL_CHECK", Catch::ResultWas::ExplicitFailure, Catch::ResultDisposition::ContinueOnFailure, __VA_ARGS__ )`
- `#define SUCCEED(...) INTERNAL_CATCH_MSG( "SUCCEED", Catch::ResultWas::Ok, Catch::ResultDisposition::ContinueOnFailure, __VA_ARGS__ )`
- `#define STATIC_REQUIRE(...) static_assert( __VA_ARGS__, #__VA_ARGS__ ); SUCCEED( #__VA_ARGS__ )`
- `#define STATIC_REQUIRE_FALSE(...) static_assert( !(__VA_ARGS__), "!(" #__VA_ARGS__ ")" ); SUCCEED( "!(" #__VA_ARGS__ ")" )`
- `#define STATIC_CHECK(...) static_assert( __VA_ARGS__, #__VA_ARGS__ ); SUCCEED( #__VA_ARGS__ )`
- `#define STATIC_CHECK_FALSE(...) static_assert( !(__VA_ARGS__), "!(" #__VA_ARGS__ ")" ); SUCCEED( "!(" #__VA_ARGS__ ")" )`
- `#define SCENARIO(...) TEST_CASE( "Scenario: " __VA_ARGS__ )`
- `#define SCENARIO_METHOD(className, ...) INTERNAL_CATCH_TEST_CASE_METHOD( className, "Scenario: " __VA_ARGS__ )`
- `#define GIVEN(desc) INTERNAL_CATCH_DYNAMIC_SECTION( " Given: " << desc )`
- `#define AND_GIVEN(desc) INTERNAL_CATCH_DYNAMIC_SECTION( " And given: " << desc )`
- `#define WHEN(desc) INTERNAL_CATCH_DYNAMIC_SECTION( " When: " << desc )`

- **#define AND\_WHEN**(desc) INTERNAL\_CATCH\_DYNAMIC\_SECTION( " And when: " << desc )
- **#define THEN**(desc) INTERNAL\_CATCH\_DYNAMIC\_SECTION( " Then: " << desc )
- **#define AND\_THEN**(desc) INTERNAL\_CATCH\_DYNAMIC\_SECTION( " And: " << desc )
- **#define CATCH\_TEMPLATE\_TEST\_REGISTRY\_HPP\_INCLUDED**
- **#define CATCH\_PREPROCESSOR\_HPP\_INCLUDED**
- **#define CATCH\_RECURSION\_LEVEL0**(...) \_\_VA\_ARGS\_\_
- **#define CATCH\_RECURSION\_LEVEL1**(...) CATCH\_RECURSION\_LEVEL0(CATCH\_RECURSION\_↵  
LEVEL0(CATCH\_RECURSION\_LEVEL0(\_\_VA\_ARGS\_\_)))
- **#define CATCH\_RECURSION\_LEVEL2**(...) CATCH\_RECURSION\_LEVEL1(CATCH\_RECURSION\_↵  
LEVEL1(CATCH\_RECURSION\_LEVEL1(\_\_VA\_ARGS\_\_)))
- **#define CATCH\_RECURSION\_LEVEL3**(...) CATCH\_RECURSION\_LEVEL2(CATCH\_RECURSION\_↵  
LEVEL2(CATCH\_RECURSION\_LEVEL2(\_\_VA\_ARGS\_\_)))
- **#define CATCH\_RECURSION\_LEVEL4**(...) CATCH\_RECURSION\_LEVEL3(CATCH\_RECURSION\_↵  
LEVEL3(CATCH\_RECURSION\_LEVEL3(\_\_VA\_ARGS\_\_)))
- **#define CATCH\_RECURSION\_LEVEL5**(...) CATCH\_RECURSION\_LEVEL4(CATCH\_RECURSION\_↵  
LEVEL4(CATCH\_RECURSION\_LEVEL4(\_\_VA\_ARGS\_\_)))
- **#define CATCH\_RECURSE**(...) CATCH\_RECURSION\_LEVEL5(\_\_VA\_ARGS\_\_)
- **#define CATCH\_REC\_END**(...)
- **#define CATCH\_REC\_OUT**
- **#define CATCH\_EMPTY**()
- **#define CATCH\_DEFER**(id) id CATCH\_EMPTY()
- **#define CATCH\_REC\_GET\_END2**() 0, CATCH\_REC\_END
- **#define CATCH\_REC\_GET\_END1**(...) CATCH\_REC\_GET\_END2
- **#define CATCH\_REC\_GET\_END**(...) CATCH\_REC\_GET\_END1
- **#define CATCH\_REC\_NEXT0**(test, next, ...) next CATCH\_REC\_OUT
- **#define CATCH\_REC\_NEXT1**(test, next) CATCH\_DEFER ( CATCH\_REC\_NEXT0 ) ( test, next, 0)
- **#define CATCH\_REC\_NEXT**(test, next) CATCH\_REC\_NEXT1(CATCH\_REC\_GET\_END test, next)
- **#define CATCH\_REC\_LIST0**(f, x, peek, ...) , f(x) CATCH\_DEFER ( CATCH\_REC\_NEXT(peek, CATCH\_↵  
REC\_LIST1) ) ( f, peek, \_\_VA\_ARGS\_\_ )
- **#define CATCH\_REC\_LIST1**(f, x, peek, ...) , f(x) CATCH\_DEFER ( CATCH\_REC\_NEXT(peek, CATCH\_↵  
REC\_LIST0) ) ( f, peek, \_\_VA\_ARGS\_\_ )
- **#define CATCH\_REC\_LIST2**(f, x, peek, ...) f(x) CATCH\_DEFER ( CATCH\_REC\_NEXT(peek, CATCH\_↵  
REC\_LIST1) ) ( f, peek, \_\_VA\_ARGS\_\_ )
- **#define CATCH\_REC\_LIST0\_UD**(f, userdata, x, peek, ...) , f(userdata, x) CATCH\_DEFER ( CATCH\_REC\_↵  
NEXT(peek, CATCH\_REC\_LIST1\_UD) ) ( f, userdata, peek, \_\_VA\_ARGS\_\_ )
- **#define CATCH\_REC\_LIST1\_UD**(f, userdata, x, peek, ...) , f(userdata, x) CATCH\_DEFER ( CATCH\_REC\_↵  
NEXT(peek, CATCH\_REC\_LIST0\_UD) ) ( f, userdata, peek, \_\_VA\_ARGS\_\_ )
- **#define CATCH\_REC\_LIST2\_UD**(f, userdata, x, peek, ...) f(userdata, x) CATCH\_DEFER ( CATCH\_REC\_↵  
NEXT(peek, CATCH\_REC\_LIST1\_UD) ) ( f, userdata, peek, \_\_VA\_ARGS\_\_ )
- **#define CATCH\_REC\_LIST\_UD**(f, userdata, ...) CATCH\_RECURSE(CATCH\_REC\_LIST2\_UD(f, userdata,  
\_\_VA\_ARGS\_\_, ()(), ()(), ()(), 0))
- **#define CATCH\_REC\_LIST**(f, ...) CATCH\_RECURSE(CATCH\_REC\_LIST2(f, \_\_VA\_ARGS\_\_, ()(), ()(),  
()(), 0))
- **#define INTERNAL\_CATCH\_STRINGIZE**(...) INTERNAL\_CATCH\_STRINGIZE2(\_\_VA\_ARGS\_\_)
- **#define INTERNAL\_CATCH\_STRINGIZE2**(...) # \_\_VA\_ARGS\_\_
- **#define INTERNAL\_CATCH\_STRINGIZE\_WITHOUT\_PARENS**(param) INTERNAL\_CATCH\_STRINGIZE(INTERNAL\_↵  
CATCH\_REMOVE\_PARENS(param))
- **#define INTERNAL\_CATCH\_MAKE\_NAMESPACE2**(...) ns\_## \_\_VA\_ARGS\_\_
- **#define INTERNAL\_CATCH\_MAKE\_NAMESPACE**(name) INTERNAL\_CATCH\_MAKE\_NAMESPACE2(name)
- **#define INTERNAL\_CATCH\_MAKE\_TYPE\_LIST2**(...) decltype(get\_wrapper<INTERNAL\_CATCH\_↵  
REMOVE\_PARENS\_GEN(\_\_VA\_ARGS\_\_)>())
- **#define INTERNAL\_CATCH\_MAKE\_TYPE\_LIST**(...) INTERNAL\_CATCH\_MAKE\_TYPE\_LIST2(INTERNAL\_↵  
CATCH\_REMOVE\_PARENS(\_\_VA\_ARGS\_\_))
- **#define INTERNAL\_CATCH\_MAKE\_TYPE\_LISTS\_FROM\_TYPES**(...) CATCH\_REC\_LIST(INTERNAL\_↵  
CATCH\_MAKE\_TYPE\_LIST, \_\_VA\_ARGS\_\_)

- `#define INTERNAL_CATCH_REMOVE_PARENS_1_ARG(_0) INTERNAL_CATCH_REMOVE_PARENS(↵  
_0)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_2_ARG(_0, _1) INTERNAL_CATCH_REMOVE_↵  
PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_1_ARG(_1)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_3_ARG(_0, _1, _2) INTERNAL_CATCH_REMOVE_↵  
PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_2_ARG(_1, _2)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_4_ARG(_0, _1, _2, _3) INTERNAL_CATCH_REMOVE_↵  
_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_3_ARG(_1, _2, _3)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_5_ARG(_0, _1, _2, _3, _4) INTERNAL_CATCH_↵  
REMOVE_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_4_ARG(_1, _2, _3, _4)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_6_ARG(_0, _1, _2, _3, _4, _5) INTERNAL_CATCH_↵  
REMOVE_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_5_ARG(_1, _2, _3, _4, _5)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_7_ARG(_0, _1, _2, _3, _4, _5, _6) INTERNAL_CATCH_↵  
_REMOVE_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_6_ARG(_1, _2, _3, _4, _5, _6)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_8_ARG(_0, _1, _2, _3, _4, _5, _6, _7) INTERNAL_↵  
CATCH_REMOVE_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_7_ARG(_1, _2, _3, _4, _5, _6,  
_7)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_9_ARG(_0, _1, _2, _3, _4, _5, _6, _7, _8) INTERNAL_↵  
_CATCH_REMOVE_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_8_ARG(_1, _2, _3, _4, _5, _6,  
_7, _8)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_10_ARG(_0, _1, _2, _3, _4, _5, _6, _7, _8, _↵  
9) INTERNAL_CATCH_REMOVE_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_9_ARG(_1, _↵  
2, _3, _4, _5, _6, _7, _8, _9)`
- `#define INTERNAL_CATCH_REMOVE_PARENS_11_ARG(_0, _1, _2, _3, _4, _5, _6, _7, _8, _9, _↵  
10) INTERNAL_CATCH_REMOVE_PARENS(_0), INTERNAL_CATCH_REMOVE_PARENS_10_ARG(_1, _↵  
2, _3, _4, _5, _6, _7, _8, _9, _10)`
- `#define INTERNAL_CATCH_VA_NARGS_IMPL(_0, _1, _2, _3, _4, _5, _6, _7, _8, _9, _10, N, ...) N`
- `#define INTERNAL_CATCH_TYPE_GEN`
- `#define INTERNAL_CATCH_NTTP_1(signature, ...)`
- `#define INTERNAL_CATCH_DECLARE_SIG_TEST0(TestName)`
- `#define INTERNAL_CATCH_DECLARE_SIG_TEST1(TestName, signature)`
- `#define INTERNAL_CATCH_DECLARE_SIG_TEST_X(TestName, signature, ...)`
- `#define INTERNAL_CATCH_DEFINE_SIG_TEST0(TestName)`
- `#define INTERNAL_CATCH_DEFINE_SIG_TEST1(TestName, signature)`
- `#define INTERNAL_CATCH_DEFINE_SIG_TEST_X(TestName, signature, ...)`
- `#define INTERNAL_CATCH_NTTP_REGISTER0(TestFunc, signature)`
- `#define INTERNAL_CATCH_NTTP_REGISTER(TestFunc, signature, ...)`
- `#define INTERNAL_CATCH_NTTP_REGISTER_METHOD0(TestName, signature, ...)`
- `#define INTERNAL_CATCH_NTTP_REGISTER_METHOD(TestName, signature, ...)`
- `#define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD0(TestName, ClassName)`
- `#define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD1(TestName, ClassName, signature)`
- `#define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD_X(TestName, ClassName, signature, ...)`
- `#define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD0(TestName)`
- `#define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD1(TestName, signature)`
- `#define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD_X(TestName, signature, ...)`
- `#define INTERNAL_CATCH_NTTP_0`
- `#define INTERNAL_CATCH_NTTP_GEN(...) INTERNAL_CATCH_VA_NARGS_IMPL(__VA_ARGS_↵  
__, INTERNAL_CATCH_NTTP_1(__VA_ARGS__), INTERNAL_CATCH_NTTP_1(__VA_ARGS__),  
INTERNAL_CATCH_NTTP_1(__VA_ARGS__), INTERNAL_CATCH_NTTP_1(__VA_ARGS__), INTERNAL_↵  
_CATCH_NTTP_1(__VA_ARGS__), INTERNAL_CATCH_NTTP_1(__VA_ARGS__), INTERNAL_CATCH_↵  
_NTTP_1(__VA_ARGS__), INTERNAL_CATCH_NTTP_1(__VA_ARGS__), INTERNAL_CATCH_NTTP_1(  
__VA_ARGS__),INTERNAL_CATCH_NTTP_1(__VA_ARGS__),INTERNAL_CATCH_NTTP_0)`



- **#define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_SIG**(Name, Tags, Signature, ...) INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_2( INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), Name, Tags, Signature, \_\_VA\_ARGS\_\_ )
- **#define INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE2**(TestName, TestFuncName, Name, Tags, Signature, TmplTypes, TypesList)
- **#define INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE**(Name, Tags, ...) INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE2(INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), Name, Tags, typename T, \_\_VA\_ARGS\_\_ )
- **#define INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE\_SIG**(Name, Tags, Signature, ...) INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE2(INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), Name, Tags, Signature, \_\_VA\_ARGS\_\_ )
- **#define INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_2**(TestName, TestFunc, Name, Tags, TmplList)
- **#define INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE**(Name, Tags, TmplList) INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_2( INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), Name, Tags, TmplList )
- **#define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_2**(TestNameClass, TestName, ClassName, Name, Tags, Signature, ...)
- **#define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD**(ClassName, Name, Tags, ...) INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_2( INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_CLASS\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), ClassName, Name, Tags, typename T, \_\_VA\_ARGS\_\_ )
- **#define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_SIG**(ClassName, Name, Tags, Signature, ...) INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_2( INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_CLASS\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), ClassName, Name, Tags, Signature, \_\_VA\_ARGS\_\_ )
- **#define INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE\_METHOD\_2**(TestNameClass, TestName, ClassName, Name, Tags, Signature, TmplTypes, TypesList)
- **#define INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE\_METHOD**(ClassName, Name, Tags, ...) INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE\_METHOD\_2( INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), ClassName, Name, Tags, typename T, \_\_VA\_ARGS\_\_ )
- **#define INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE\_METHOD\_SIG**(ClassName, Name, Tags, Signature, ...) INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE\_METHOD\_2( INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), ClassName, Name, Tags, Signature, \_\_VA\_ARGS\_\_ )
- **#define INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_METHOD\_2**(TestNameClass, TestName, ClassName, Name, Tags, TmplList)
- **#define INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_METHOD**(ClassName, Name, Tags, TmplList) INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_METHOD\_2( INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), INTERNAL\_CATCH\_UNIQUE\_NAME( CATCH2\_INTERNAL\_TEMPLATE\_TEST\_ ), ClassName, Name, Tags, TmplList )
- **#define TEMPLATE\_TEST\_CASE**(...) INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE( \_\_VA\_ARGS\_\_ )
- **#define TEMPLATE\_TEST\_CASE\_SIG**(...) INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_SIG( \_\_VA\_ARGS\_\_ )
- **#define TEMPLATE\_TEST\_CASE\_METHOD**(className, ...) INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD( className, \_\_VA\_ARGS\_\_ )
- **#define TEMPLATE\_TEST\_CASE\_METHOD\_SIG**(className, ...) INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_SIG( className, \_\_VA\_ARGS\_\_ )
- **#define TEMPLATE\_PRODUCT\_TEST\_CASE**(...) INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE( \_\_VA\_ARGS\_\_ )

- `#define TEMPLATE_PRODUCT_TEST_CASE_SIG(...) INTERNAL_CATCH_TEMPLATE_PRODUCT_↵  
TEST_CASE_SIG( __VA_ARGS__ )`
- `#define TEMPLATE_PRODUCT_TEST_CASE_METHOD(className, ...) INTERNAL_CATCH_↵  
TEMPLATE_PRODUCT_TEST_CASE_METHOD( className, __VA_ARGS__ )`
- `#define TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(className, ...) INTERNAL_CATCH_↵  
TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG( className, __VA_ARGS__ )`
- `#define TEMPLATE_LIST_TEST_CASE(...) INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE(↵  
__VA_ARGS__ )`
- `#define TEMPLATE_LIST_TEST_CASE_METHOD(className, ...) INTERNAL_CATCH_TEMPLATE_↵  
LIST_TEST_CASE_METHOD( className, __VA_ARGS__ )`
- `#define CATCH_TEST_CASE_INFO_HPP_INCLUDED`
- `#define CATCH_TRANSLATE_EXCEPTION_HPP_INCLUDED`
- `#define CATCH_INTERFACES_EXCEPTION_HPP_INCLUDED`
- `#define INTERNAL_CATCH_TRANSLATE_EXCEPTION2(translatorName, signature)`
- `#define INTERNAL_CATCH_TRANSLATE_EXCEPTION(signature) INTERNAL_CATCH_TRANSLATE_↵  
EXCEPTION2( INTERNAL_CATCH_UNIQUE_NAME( catch_internal_ExceptionTranslator ), signature )`
- `#define CATCH_TRANSLATE_EXCEPTION(signature) INTERNAL_CATCH_TRANSLATE_EXCEPTION(↵  
signature )`
- `#define CATCH_VERSION_HPP_INCLUDED`
- `#define CATCH_VERSION_MACROS_HPP_INCLUDED`
- `#define CATCH_VERSION_MAJOR 3`
- `#define CATCH_VERSION_MINOR 0`
- `#define CATCH_VERSION_PATCH 1`
- `#define CATCH_GENERATORS_ALL_HPP_INCLUDED`
- `#define CATCH_GENERATOR_EXCEPTION_HPP_INCLUDED`
- `#define CATCH_GENERATORS_HPP_INCLUDED`
- `#define CATCH_INTERFACES_GENERATORTRACKER_HPP_INCLUDED`
- `#define GENERATE(...)`
- `#define GENERATE_COPY(...)`
- `#define GENERATE_REF(...)`
- `#define CATCH_GENERATORS_ADAPTERS_HPP_INCLUDED`
- `#define CATCH_GENERATORS_RANDOM_HPP_INCLUDED`
- `#define CATCH_RANDOM_NUMBER_GENERATOR_HPP_INCLUDED`
- `#define CATCH_GENERATORS_RANGE_HPP_INCLUDED`
- `#define CATCH_INTERFACES_ALL_HPP_INCLUDED`
- `#define CATCH_INTERFACES_REPORTER_FACTORY_HPP_INCLUDED`
- `#define CATCH_INTERFACES_REPORTER_REGISTRY_HPP_INCLUDED`
- `#define CATCH_CASE_INSENSITIVE_COMPARISONS_HPP_INCLUDED`
- `#define CATCH_INTERFACES_TAG_ALIAS_REGISTRY_HPP_INCLUDED`
- `#define CATCH_CONFIG_ANDROID_LOGWRITE_HPP_INCLUDED`
- `#define CATCH_CONFIG_UNCAUGHT_EXCEPTIONS_HPP_INCLUDED`
- `#define CATCH_CONFIG_CONSOLE_WIDTH_HPP_INCLUDED`
- `#define CATCH_CONFIG_CONSOLE_WIDTH 80`
- `#define CATCH_CONTAINER_NONMEMBERS_HPP_INCLUDED`
- `#define CATCH_CONFIG_POLYFILL_NONMEMBER_CONTAINER_ACCESS`
- `#define CATCH_DEBUG_CONSOLE_HPP_INCLUDED`
- `#define CATCH_DEBUGGER_HPP_INCLUDED`
- `#define CATCH_BREAK_INTO_DEBUGGER() []{}()`
- `#define CATCH_ENFORCE_HPP_INCLUDED`
- `#define CATCH_MAKE_MSG(...) (Catch::ReusableStringStream() << __VA_ARGS__).str()`
- `#define CATCH_INTERNAL_ERROR(...) Catch::throw_logic_error(CATCH_MAKE_MSG( CATCH_↵  
INTERNAL_LINEINFO << ": Internal Catch2 error: " << __VA_ARGS__ ))`
- `#define CATCH_ERROR(...) Catch::throw_domain_error(CATCH_MAKE_MSG( __VA_ARGS__ ))`
- `#define CATCH_RUNTIME_ERROR(...) Catch::throw_runtime_error(CATCH_MAKE_MSG( __VA_ARGS_↵  
__ ))`

- `#define CATCH_ENFORCE(condition, ...) do{ if( !(condition) ) CATCH_ERROR( __VA_ARGS__ ); }`  
`while(false)`
- `#define CATCH_ENUM_VALUES_REGISTRY_HPP_INCLUDED`
- `#define CATCH_ERRNO_GUARD_HPP_INCLUDED`
- `#define CATCH_EXCEPTION_TRANSLATOR_REGISTRY_HPP_INCLUDED`
- `#define CATCH_FATAL_CONDITION_HANDLER_HPP_INCLUDED`
- `#define CATCH_FLOATING_POINT_HELPERS_HPP_INCLUDED`
- `#define CATCH_POLYFILLS_HPP_INCLUDED`
- `#define CATCH_ISTREAM_HPP_INCLUDED`
- `#define CATCH_LEAK_DETECTOR_HPP_INCLUDED`
- `#define CATCH_LIST_HPP_INCLUDED`
- `#define CATCH_OUTPUT_REDIRECT_HPP_INCLUDED`
- `#define CATCH_REPORTER_REGISTRY_HPP_INCLUDED`
- `#define CATCH_RUN_CONTEXT_HPP_INCLUDED`
- `#define CATCH_TEST_CASE_TRACKER_HPP_INCLUDED`
- `#define CATCH_SHARDING_HPP_INCLUDED`
- `#define CATCH_SINGLETONS_HPP_INCLUDED`
- `#define CATCH_STARTUP_EXCEPTION_REGISTRY_HPP_INCLUDED`
- `#define CATCH_STDSTREAMS_HPP_INCLUDED`
- `#define CATCH_STRING_MANIP_HPP_INCLUDED`
- `#define CATCH_TAG_ALIAS_REGISTRY_HPP_INCLUDED`
- `#define CATCH_TEST_CASE_INFO_HASHER_HPP_INCLUDED`
- `#define CATCH_TEST_CASE_REGISTRY_IMPL_HPP_INCLUDED`
- `#define CATCH_TEST_SPEC_PARSER_HPP_INCLUDED`
- `#define CATCH_TEXTFLOW_HPP_INCLUDED`
- `#define CATCH_TO_STRING_HPP_INCLUDED`
- `#define CATCH_UNCAUGHT_EXCEPTIONS_HPP_INCLUDED`
- `#define CATCH_WINDOWS_H_PROXY_HPP_INCLUDED`
- `#define CATCH_XMLWRITER_HPP_INCLUDED`
- `#define CATCH_MATCHERS_ALL_HPP_INCLUDED`
- `#define CATCH_MATCHERS_HPP_INCLUDED`
- `#define CATCH_MATCHERS_IMPL_HPP_INCLUDED`
- `#define INTERNAL_CHECK_THAT(macroName, matcher, resultDisposition, arg)`
- `#define INTERNAL_CATCH_THROWS_MATCHES(macroName, exceptionType, resultDisposition, matcher,`  
`...)`
- `#define REQUIRE_THROWS_WITH(expr, matcher) INTERNAL_CATCH_THROWS_STR_MATCHES(`  
`"REQUIRE_THROWS_WITH", Catch::ResultDisposition::Normal, matcher, expr )`
- `#define REQUIRE_THROWS_MATCHES(expr, exceptionType, matcher) INTERNAL_CATCH_THROWS_↵`  
`MATCHES( "REQUIRE_THROWS_MATCHES", exceptionType, Catch::ResultDisposition::Normal, matcher,`  
`expr )`
- `#define CHECK_THROWS_WITH(expr, matcher) INTERNAL_CATCH_THROWS_STR_MATCHES(`  
`"CHECK_THROWS_WITH", Catch::ResultDisposition::ContinueOnFailure, matcher, expr )`
- `#define CHECK_THROWS_MATCHES(expr, exceptionType, matcher) INTERNAL_CATCH_THROWS_↵`  
`MATCHES( "CHECK_THROWS_MATCHES", exceptionType, Catch::ResultDisposition::ContinueOnFailure,`  
`matcher, expr )`
- `#define CHECK_THAT(arg, matcher) INTERNAL_CHECK_THAT( "CHECK_THAT", matcher, Catch::↵`  
`ResultDisposition::ContinueOnFailure, arg )`
- `#define REQUIRE_THAT(arg, matcher) INTERNAL_CHECK_THAT( "REQUIRE_THAT", matcher, Catch::↵`  
`ResultDisposition::Normal, arg )`
- `#define CATCH_MATCHERS_CONTAINER_PROPERTIES_HPP_INCLUDED`
- `#define CATCH_MATCHERS_TEMPLATED_HPP_INCLUDED`
- `#define CATCH_MATCHERS_CONTAINS_HPP_INCLUDED`
- `#define CATCH_MATCHERS_EXCEPTION_HPP_INCLUDED`
- `#define CATCH_MATCHERS_FLOATING_POINT_HPP_INCLUDED`
- `#define CATCH_MATCHERS_PREDICATE_HPP_INCLUDED`

- `#define CATCH_MATCHERS_QUANTIFIERS_HPP_INCLUDED`
- `#define CATCH_MATCHERS_STRING_HPP_INCLUDED`
- `#define CATCH_MATCHERS_VECTOR_HPP_INCLUDED`
- `#define CATCH_REPORTERS_ALL_HPP_INCLUDED`
- `#define CATCH_REPORTER_AUTOMAKE_HPP_INCLUDED`
- `#define CATCH_REPORTER_STREAMING_BASE_HPP_INCLUDED`
- `#define CATCH_REPORTER_COMMON_BASE_HPP_INCLUDED`
- `#define CATCH_REPORTER_COMPACT_HPP_INCLUDED`
- `#define CATCH_REPORTER_CONSOLE_HPP_INCLUDED`
- `#define CATCH_REPORTER_CUMULATIVE_BASE_HPP_INCLUDED`
- `#define CATCH_REPORTER_EVENT_LISTENER_HPP_INCLUDED`
- `#define CATCH_REPORTER_HELPERS_HPP_INCLUDED`
- `#define CATCH_REPORTER_JUNIT_HPP_INCLUDED`
- `#define CATCH_REPORTER_MULTI_HPP_INCLUDED`
- `#define CATCH_REPORTER_REGISTRARS_HPP_INCLUDED`
- `#define CATCH_REGISTER_REPORTER(name, reporterType)`
- `#define CATCH_REGISTER_LISTENER(listenerType)`
- `#define CATCH_REPORTER_SONARQUBE_HPP_INCLUDED`
- `#define CATCH_REPORTER_TAP_HPP_INCLUDED`
- `#define CATCH_REPORTER_TEAMCITY_HPP_INCLUDED`
- `#define CATCH_REPORTER_XML_HPP_INCLUDED`

## Typedefs

- using **Catch::IEventListenerPtr** = Detail::unique\_ptr< IEventListener >
- template<typename Clock >  
using **Catch::Benchmark::ClockDuration** = typename Clock::duration
- template<typename Clock >  
using **Catch::Benchmark::FloatDuration** = std::chrono::duration< double, typename Clock::period >
- template<typename Clock >  
using **Catch::Benchmark::TimePoint** = typename Clock::time\_point
- using **Catch::Benchmark::default\_clock** = std::chrono::steady\_clock
- using **Catch::Benchmark::fp\_seconds** = std::chrono::duration< double, std::ratio< 1 > >
- template<typename Func, typename... U>  
using **Catch::FunctionReturnType** = std::remove\_reference\_t< std::remove\_cv\_t< std::result\_of\_t< Func(U...) > >
- using **Catch::IReporterFactoryPtr** = Detail::unique\_ptr< IReporterFactory >
- template<typename T >  
using **Catch::Benchmark::Detail::CompleteType\_t** = typename CompleteType< T >::type
- template<typename Clock, typename Func, typename... Args>  
using **Catch::Benchmark::TimingOf** = Timing< ClockDuration< Clock >, Detail::CompleteType\_t< FunctionReturnType< Func, Args... > >
- template<typename Clock, typename Fun >  
using **Catch::Benchmark::Detail::run\_for\_at\_least\_argument\_t** = std::conditional\_t< is\_callable< Fun(Chronometer)>::value, Chronometer, int >
- using **Catch::Benchmark::Detail::sample** = std::vector< double >
- template<typename T >  
using **Catch::Benchmark::storage\_for** = Detail::ObjectStorage< T, true >
- template<typename T >  
using **Catch::Benchmark::destructable\_object** = Detail::ObjectStorage< T, false >
- template<typename... Ts>  
using **Catch::Detail::void\_t** = typename make\_void< Ts... >::type
- using **Catch::Clara::Detail::Result** = BasicResult< void >
- using **Catch::Clara::Detail::ParserResult** = BasicResult< ParseResultType >



- using **Catch::Clara::Detail::InternalParseResult** = BasicResult< ParseState >
- using **Catch::exceptionTranslateFunction** = std::string(\*)()
- using **Catch::ExceptionTranslators** = std::vector< Detail::unique\_ptr< IExceptionTranslator const > >
- using **Catch::Generators::GeneratorBasePtr** = [Catch::Detail::unique\\_ptr](#)< GeneratorUntypedBase >
- template<typename T >  
using **Catch::Generators::GeneratorPtr** = [Catch::Detail::unique\\_ptr](#)< IGenerator< T > >
- using **Catch::TestCaseTracking::ITrackerPtr** = [Catch::Detail::unique\\_ptr](#)< ITracker >
- using **Catch::StringMatcher** = Matchers::MatcherBase< std::string >
- template<typename T >  
using **Catch::Matchers::Detail::is\_generic\_matcher** = std::is\_base\_of< [Catch::Matchers::MatcherGenericBase](#), std::remove\_cv\_t< std::remove\_reference\_t< T > > >
- template<typename... Ts>  
using **Catch::Matchers::Detail::are\_generic\_matchers** = conjunction< is\_generic\_matcher< Ts >... >
- template<typename T >  
using **Catch::Matchers::Detail::is\_matcher** = std::is\_base\_of< [Catch::Matchers::MatcherUntypedBase](#), std::remove\_cv\_t< std::remove\_reference\_t< T > > >

## Enumerations

- enum class **Verbosity** { **Quiet** = 0 , **Normal** , **High** }
  - enum class **ShowDurations** { **DefaultForReporter** , **Always** , **Never** }
  - enum class **TestRunOrder** { **Declared** , **LexicographicallySorted** , **Randomized** }
  - enum class [Catch::ColourMode](#) : std::uint8\_t { [PlatformDefault](#) , [ANSI](#) , [Win32](#) , [None](#) }
  - enum class **CaseSensitive** { **Yes** , **No** }
  - enum class [Catch::GenerateFrom](#) { **Time** , **RandomDevice** , **Default** }
  - enum class **ParseResultType** { **Matched** , **NoMatch** , **ShortCircuitAll** , **ShortCircuitSame** }
  - enum class **TokenType** { **Option** , **Argument** }
  - enum class [Catch::Clara::Detail::ResultType](#) { [Ok](#) , [LogicError](#) , [RuntimeError](#) }
- Denotes type of a parsing result.*
- enum class **Optionality** { **Optional** , **Required** }
  - enum class **TestCaseProperties** : uint8\_t {  
**None** = 0 , **IsHidden** = 1 << 1 , **ShouldFail** = 1 << 2 , **MayFail** = 1 << 3 ,  
**Throws** = 1 << 4 , **NonPortable** = 1 << 5 , **Benchmark** = 1 << 6 }
  - enum class **XmlFormatting** { **None** = 0x00 , **Indent** = 0x01 , **Newline** = 0x02 }

## Functions

- constexpr auto **Catch::operator""\_sr** (char const \*rawChars, std::size\_t size) noexcept -> StringRef
- constexpr auto **operator""\_catch\_sr** (char const \*rawChars, std::size\_t size) noexcept -> [Catch::StringRef](#)
- IMutableContext & **Catch::getCurrentMutableContext** ()
- IContext & **Catch::getCurrentContext** ()
- void **Catch::cleanUpContext** ()
- SimplePcg32 & **Catch::sharedRng** ()
- bool **Catch::isOk** (ResultWas::OfType resultType)
- bool **Catch::isJustInfo** (int flags)
- ResultDisposition::Flags **Catch::operator|** (ResultDisposition::Flags lhs, ResultDisposition::Flags rhs)
- bool **Catch::shouldContinueOnFailure** (int flags)
- bool **Catch::isFalseTest** (int flags)
- bool **Catch::shouldSuppressFailure** (int flags)
- IResultCapture & **Catch::getResultCapture** ()
- template<typename T, typename... Args>  
unique\_ptr< T > **Catch::Detail::make\_unique** (Args &&... args)

- `template<typename T >`  
`void Catch::Benchmark::deoptimize_value (T &&x)`
- `template<typename Fn , typename... Args>`  
`auto Catch::Benchmark::invoke_deoptimized (Fn &&fn, Args &&... args) -> std::enable_if_t<!std::is_↵`  
`same< void, decltype(fn(args...))>::value >`
- `IRegistryHub const & Catch::getRegistryHub ()`
- `IMutableRegistryHub & Catch::getMutableRegistryHub ()`
- `void Catch::cleanUp ()`
- `std::string Catch::translateActiveException ()`
- `template<typename Fun , typename... Args>`  
`CompleteType_t< FunctionReturnType< Fun, Args... > > Catch::Benchmark::Detail::complete_invoke`  
`(Fun &&fun, Args &&... args)`
- `template<typename Fun >`  
`Detail::CompleteType_t< FunctionReturnType< Fun > > Catch::Benchmark::user_code (Fun &&fun)`
- `template<typename Fun >`  
`repeater< std::decay_t< Fun > > Catch::Benchmark::Detail::repeat (Fun &&fun)`
- `template<typename Clock , typename Fun , typename... Args>`  
`TimingOf< Clock, Fun, Args... > Catch::Benchmark::Detail::measure (Fun &&fun, Args &&... args)`
- `template<typename Clock , typename Fun >`  
`TimingOf< Clock, Fun, int > Catch::Benchmark::Detail::measure_one (Fun &&fun, int iters, std::false_↵`  
`type)`
- `template<typename Clock , typename Fun >`  
`TimingOf< Clock, Fun, Chronometer > Catch::Benchmark::Detail::measure_one (Fun &&fun, int iters,`  
`std::true_type)`
- `void Catch::Benchmark::Detail::throw_optimized_away_error ()`
- `template<typename Clock , typename Fun >`  
`TimingOf< Clock, Fun, run_for_at_least_argument_t< Clock, Fun > > Catch::Benchmark::Detail::run_↵`  
`for_at_least (ClockDuration< Clock > how_long, const int initial_iterations, Fun &&fun)`
- `bool Catch::Benchmark::Detail::directCompare (double lhs, double rhs)`
- `double Catch::Benchmark::Detail::weighted_average_quantile (int k, int q, std::vector< double >↵`  
`::iterator first, std::vector< double >::iterator last)`
- `template<typename Iterator >`  
`OutlierClassification Catch::Benchmark::Detail::classify_outliers (Iterator first, Iterator last)`
- `template<typename Iterator >`  
`double Catch::Benchmark::Detail::mean (Iterator first, Iterator last)`
- `template<typename Estimator , typename Iterator >`  
`sample Catch::Benchmark::Detail::jackknife (Estimator &&estimator, Iterator first, Iterator last)`
- `double Catch::Benchmark::Detail::normal_cdf (double x)`
- `double Catch::Benchmark::Detail::erfc_inv (double x)`
- `double Catch::Benchmark::Detail::normal_quantile (double p)`
- `template<typename Iterator , typename Estimator >`  
`Estimate< double > Catch::Benchmark::Detail::bootstrap (double confidence_level, Iterator first, Iterator`  
`last, sample const &resample, Estimator &&estimator)`
- `double Catch::Benchmark::Detail::outlier_variance (Estimate< double > mean, Estimate< double >`  
`stddev, int n)`
- `bootstrap_analysis Catch::Benchmark::Detail::analyse_samples (double confidence_level, unsigned int`  
`n_resamples, std::vector< double >::iterator first, std::vector< double >::iterator last)`
- `template<typename Clock >`  
`std::vector< double > Catch::Benchmark::Detail::resolution (int k)`
- `template<typename Clock >`  
`int Catch::Benchmark::Detail::warmup ()`
- `template<typename Clock >`  
`EnvironmentEstimate< FloatDuration< Clock > > Catch::Benchmark::Detail::estimate_clock_↵`  
`resolution (int iterations)`
- `template<typename Clock >`  
`EnvironmentEstimate< FloatDuration< Clock > > Catch::Benchmark::Detail::estimate_clock_cost`  
`(FloatDuration< Clock > resolution)`

- `template<typename Clock >`  
`Environment< FloatDuration< Clock > > Catch::Benchmark::Detail::measure_environment ()`
- `template<typename Duration , typename Iterator >`  
`SampleAnalysis< Duration > Catch::Benchmark::Detail::analyse (const IConfig &cfg, Environment< Duration >, Iterator first, Iterator last)`
- `std::ostream & operator<< (std::ostream &, Catch\_global\_namespace\_dummy)`
- `std::string Catch::Detail::convertIntoString (StringRef string, bool escapeInvisibles)`  
*Encases `string` in quotes, and optionally escapes invisibles.*
- `std::string Catch::Detail::convertIntoString (StringRef string)`
- `std::string Catch::Detail::rawMemoryToString (const void *object, std::size_t size)`
- `template<typename T >`  
`std::string Catch::Detail::rawMemoryToString (const T &object)`
- `template<typename E >`  
`std::string Catch::Detail::convertUnknownEnumToString (E e)`
- `template<typename T >`  
`std::enable_if_t< !std::is_enum< T >::value &&!std::is_base_of< std::exception, T >::value, std::string > Catch::Detail::convertUnstreamable (T const &)`
- `template<typename T >`  
`std::enable_if_t< !std::is_enum< T >::value &&std::is_base_of< std::exception, T >::value, std::string > Catch::Detail::convertUnstreamable (T const &ex)`
- `template<typename T >`  
`std::enable_if_t< std::is_enum< T >::value, std::string > Catch::Detail::convertUnstreamable (T const &value)`
- `template<typename T >`  
`std::string Catch::Detail::stringify (const T &e)`
- `template<typename InputIterator , typename Sentinel = InputIterator>`  
`std::string Catch::Detail::rangeToString (InputIterator first, Sentinel last)`
- `template<typename Range >`  
`std::string Catch::rangeToString (Range const &range)`
- `template<typename Allocator >`  
`std::string Catch::rangeToString (std::vector< bool, Allocator > const &v)`
- Approx `Catch::literals::operator""_a` (long double val)
- Approx `Catch::literals::operator""_a` (unsigned long long val)
- `std::uint32_t Catch::generateRandomSeed (GenerateFrom from)`
- `Detail::unique_ptr< ColourImpl > Catch::makeColourImpl (ColourMode colourSelection, IStream *stream)`  
*Provides [ColourImpl](#) based on global config and target compilation platform.*
- `bool Catch::isColourImplAvailable (ColourMode colourSelection)`  
*Checks if specific colour impl has been compiled into the binary.*
- `std::vector< std::string > Catch::Detail::splitReporterSpec (StringRef reporterSpec)`  
*Splits the reporter spec into reporter name and kv-pair options.*
- `Optional< ColourMode > Catch::Detail::stringToColourMode (StringRef colourMode)`
- `Optional< ReporterSpec > Catch::parseReporterSpec (StringRef reporterSpec)`
- `template<typename T >`  
`ParserResult Catch::Clara::Detail::convertInto (std::string const &source, T &target)`
- `ParserResult Catch::Clara::Detail::convertInto (std::string const &source, std::string &target)`
- `ParserResult Catch::Clara::Detail::convertInto (std::string const &source, bool &target)`
- `template<typename ArgType , typename L >`  
`auto Catch::Clara::Detail::invokeLambda (L const &lambda, std::string const &arg) -> ParserResult`
- `Clara::Parser Catch::makeCommandLineParser (ConfigData &config)`
- `void Catch::formatReconstructedExpression (std::ostream &os, std::string const &lhs, StringRef op, std::string const &rhs)`
- `template<typename LhsT , typename RhsT >`  
`auto Catch::compareEqual (LhsT const &lhs, RhsT const &rhs) -> bool`
- `template<typename T >`  
`auto Catch::compareEqual (T *const &lhs, int rhs) -> bool`

- `template<typename T >`  
`auto Catch::compareEqual (T *const &lhs, long rhs) -> bool`
- `template<typename T >`  
`auto Catch::compareEqual (int lhs, T *const &rhs) -> bool`
- `template<typename T >`  
`auto Catch::compareEqual (long lhs, T *const &rhs) -> bool`
- `template<typename LhsT , typename RhsT >`  
`auto Catch::compareNotEqual (LhsT const &lhs, RhsT &&rhs) -> bool`
- `template<typename T >`  
`auto Catch::compareNotEqual (T *const &lhs, int rhs) -> bool`
- `template<typename T >`  
`auto Catch::compareNotEqual (T *const &lhs, long rhs) -> bool`
- `template<typename T >`  
`auto Catch::compareNotEqual (int lhs, T *const &rhs) -> bool`
- `template<typename T >`  
`auto Catch::compareNotEqual (long lhs, T *const &rhs) -> bool`
- `void Catch::handleExceptionMatchExpr (AssertionHandler &handler, std::string const &str, StringRef matcherString)`
- `bool Catch::isThrowSafe (TestCaseHandle const &testCase, IConfig const &config)`
- `bool Catch::matchTest (TestCaseHandle const &testCase, TestSpec const &testSpec, IConfig const &config)`
- `std::vector< TestCaseHandle > Catch::filterTests (std::vector< TestCaseHandle > const &testCases, TestSpec const &testSpec, IConfig const &config)`
- `std::vector< TestCaseHandle > const & Catch::getAllTestCasesSorted (IConfig const &config)`
- `Detail::unique_ptr< ITestInvoker > Catch::makeTestInvoker (void(*testAsFunction)())`
- `template<typename C >`  
`Detail::unique_ptr< ITestInvoker > Catch::makeTestInvoker (void(C::*testAsMethod)())`
- `Detail::unique_ptr< TestCaseInfo > Catch::makeTestCaseInfo (StringRef className, NameAndTags const &nameAndTags, SourceLineInfo const &lineInfo)`
- `Version const & Catch::libraryVersion ()`
- `void Catch::Generators::Detail::throw_generator_exception (char const *msg)`  
*Throws [GeneratorException](#) with the provided message.*
- `template<typename T , typename DecayedT = std::decay_t<T>>`  
`GeneratorWrapper< DecayedT > Catch::Generators::value (T &&value)`
- `template<typename T >`  
`GeneratorWrapper< T > Catch::Generators::values (std::initializer_list< T > values)`
- `template<typename... Ts>`  
`GeneratorWrapper< std::tuple< std::decay_t< Ts >... > > Catch::Generators::table (std::initializer_list< std::tuple< std::decay_t< Ts >... >> tuples)`
- `template<typename T , typename... Gs>`  
`auto Catch::Generators::makeGenerators (GeneratorWrapper< T > &&generator, Gs &&... moreGenerators) -> Generators< T >`
- `template<typename T >`  
`auto Catch::Generators::makeGenerators (GeneratorWrapper< T > &&generator) -> Generators< T >`
- `template<typename T , typename... Gs>`  
`auto Catch::Generators::makeGenerators (T &&val, Gs &&... moreGenerators) -> Generators< std::decay_t< T >>`
- `template<typename T , typename U , typename... Gs>`  
`auto Catch::Generators::makeGenerators (as< T >, U &&val, Gs &&... moreGenerators) -> Generators< T >`
- `auto Catch::Generators::acquireGeneratorTracker (StringRef generatorName, SourceLineInfo const &lineInfo) -> IGeneratorTracker &`
- `template<typename L >`  
`auto Catch::Generators::generate (StringRef generatorName, SourceLineInfo const &lineInfo, L const &generatorExpression) -> decltype(std::declval< decltype(generatorExpression())>().get())`

- `template<typename T >`  
`GeneratorWrapper< T > Catch::Generators::take (size_t target, GeneratorWrapper< T > &&generator)`
- `template<typename T, typename Predicate >`  
`GeneratorWrapper< T > Catch::Generators::filter (Predicate &&pred, GeneratorWrapper< T > &&generator)`
- `template<typename T >`  
`GeneratorWrapper< T > Catch::Generators::repeat (size_t repeats, GeneratorWrapper< T > &&generator)`
- `template<typename Func, typename U, typename T = FunctionReturnType<Func, U>>`  
`GeneratorWrapper< T > Catch::Generators::map (Func &&function, GeneratorWrapper< U > &&generator)`
- `template<typename T >`  
`GeneratorWrapper< std::vector< T > > Catch::Generators::chunk (size_t size, GeneratorWrapper< T > &&generator)`
- `std::uint32_t Catch::Generators::Detail::getSeed ()`
- `template<typename T >`  
`std::enable_if_t< std::is_integral< T >::value &&!std::is_same< T, bool >::value, GeneratorWrapper< T > > Catch::Generators::random (T a, T b)`
- `template<typename T >`  
`std::enable_if_t< std::is_floating_point< T >::value, GeneratorWrapper< T > > Catch::Generators::random (T a, T b)`
- `template<typename T >`  
`GeneratorWrapper< T > Catch::Generators::range (T const &start, T const &end, T const &step)`
- `template<typename T >`  
`GeneratorWrapper< T > Catch::Generators::range (T const &start, T const &end)`
- `template<typename InputIterator, typename InputSentinel, typename ResultType = typename std::iterator_traits<InputIterator>::value_type>`  
`GeneratorWrapper< ResultType > Catch::Generators::from_range (InputIterator from, InputSentinel to)`
- `template<typename Container, typename ResultType = typename Container::value_type>`  
`GeneratorWrapper< ResultType > Catch::Generators::from_range (Container const &cnt)`
- `template<typename Container >`  
`constexpr auto Catch::Detail::empty (Container const &cont) -> decltype(cont.empty())`
- `template<typename T, std::size_t N>`  
`constexpr bool Catch::Detail::empty (const T(&)[N]) noexcept`
- `template<typename T >`  
`constexpr bool Catch::Detail::empty (std::initializer_list< T > list) noexcept`
- `template<typename Container >`  
`constexpr auto Catch::Detail::size (Container const &cont) -> decltype(cont.size())`
- `template<typename T, std::size_t N>`  
`constexpr std::size_t Catch::Detail::size (const T(&)[N]) noexcept`
- `void Catch::writeToDebugConsole (std::string const &text)`
- `bool Catch::isDebuggerActive ()`
- `void Catch::throw_exception (std::exception const &e)`
- `void Catch::throw_logic_error (std::string const &msg)`
- `void Catch::throw_domain_error (std::string const &msg)`
- `void Catch::throw_runtime_error (std::string const &msg)`
- `Catch::Detail::unique\_ptr< EnumInfo > Catch::Detail::makeEnumInfo (StringRef enumName, StringRef allValueNames, std::vector< int > const &values)`
- `std::vector< StringRef > Catch::Detail::parseEnums (StringRef enums)`
- `bool Catch::isnan (float f)`
- `bool Catch::isnan (double d)`
- `uint32_t Catch::Detail::convertToBits (float f)`
- `uint64_t Catch::Detail::convertToBits (double d)`
- `template<typename FP >`  
`uint64_t Catch::ulpDistance (FP lhs, FP rhs)`
- `auto Catch::makeStream (std::string const &filename) -> Detail::unique_ptr< IStream >`

- bool **Catch::list** (IEventListener &reporter, Config const &config)
- void **Catch::seedRng** (IConfig const &config)
- unsigned int **Catch::rngSeed** ()
- template<typename Container >  
Container **Catch::createShard** (Container const &container, std::size\_t const shardCount, std::size\_t const shardIndex)
- void **Catch::addSingleton** (ISingleton \*singleton)
- void **Catch::cleanupSingletons** ()
- std::ostream & **Catch::cout** ()
- std::ostream & **Catch::cerr** ()
- std::ostream & **Catch::clog** ()
- bool **Catch::startsWith** (std::string const &s, std::string const &prefix)
- bool **Catch::startsWith** (StringRef s, char prefix)
- bool **Catch::endsWith** (std::string const &s, std::string const &suffix)
- bool **Catch::endsWith** (std::string const &s, char suffix)
- bool **Catch::contains** (std::string const &s, std::string const &infix)
- void **Catch::toLowerInPlace** (std::string &s)
- std::string **Catch::toLower** (std::string const &s)
- char **Catch::toLower** (char c)
- std::string **Catch::trim** (std::string const &str)  
*Returns a new string without whitespace at the start/end.*
- StringRef **Catch::trim** (StringRef ref)  
*Returns a substring of the original ref without whitespace. Beware lifetimes!*
- std::vector< StringRef > **Catch::splitStringRef** (StringRef str, char delimiter)
- bool **Catch::replaceInPlace** (std::string &str, std::string const &replaceThis, std::string const &withThis)
- std::vector< TestCaseHandle > **Catch::sortTests** (IConfig const &config, std::vector< TestCaseHandle > const &unsortedTestCases)
- void **Catch::enforceNoDuplicateTestCases** (std::vector< TestCaseHandle > const &functions)
- TestSpec **Catch::parseTestSpec** (std::string const &arg)
- Column **Catch::TextFlow::Spacer** (size\_t spaceWidth)  
*Creates a column that serves as an empty space of specific width.*
- template<typename T >  
std::string **Catch::to\_string** (T const &t)
- bool **Catch::uncaught\_exceptions** ()
- XmlFormatting **Catch::operator|** (XmlFormatting lhs, XmlFormatting rhs)
- XmlFormatting **Catch::operator&** (XmlFormatting lhs, XmlFormatting rhs)
- void **Catch::handleExceptionMatchExpr** (AssertionHandler &handler, StringMatcher const &matcher, StringRef matcherString)
- template<typename ArgT, typename MatcherT >  
auto **Catch::makeMatchExpr** (ArgT &&arg, MatcherT const &matcher, StringRef matcherString) -> MatchExpr< ArgT, MatcherT >
- template<typename ArgT >  
MatchAllOf< ArgT > **Catch::Matchers::Detail::operator&&** (MatchAllOf< ArgT > const &lhs, MatcherBase< ArgT > const &rhs)=delete
- template<typename ArgT >  
MatchAllOf< ArgT > **Catch::Matchers::Detail::operator&&** (MatcherBase< ArgT > const &lhs, MatchAllOf< ArgT > const &rhs)=delete
- template<typename ArgT >  
MatchAnyOf< ArgT > **Catch::Matchers::Detail::operator||** (MatchAnyOf< ArgT > const &lhs, MatcherBase< ArgT > const &rhs)=delete
- template<typename ArgT >  
MatchAnyOf< ArgT > **Catch::Matchers::Detail::operator||** (MatcherBase< ArgT > const &lhs, MatchAnyOf< ArgT > const &rhs)=delete
- template<typename T >  
Detail::MatchAllOf< T > **Catch::Matchers::operator&&** (MatcherBase< T > const &lhs, MatcherBase< T > const &rhs)

- `template<typename T >`  
`Detail::MatchAnyOf< T > Catch::Matchers::operator|| (MatcherBase< T > const &lhs, MatcherBase< T > const &rhs)`
- `template<typename T >`  
`Detail::MatchNotOf< T > Catch::Matchers::operator! (MatcherBase< T > const &matcher)`
- `template<std::size_t N, std::size_t M>`  
`std::array< void const *, N+M > Catch::Matchers::Detail::array_cat (std::array< void const *, N > &&lhs, std::array< void const *, M > &&rhs)`
- `template<std::size_t N>`  
`std::array< void const *, N+1 > Catch::Matchers::Detail::array_cat (std::array< void const *, N > &&lhs, void const *rhs)`
- `template<std::size_t N>`  
`std::array< void const *, N+1 > Catch::Matchers::Detail::array_cat (void const *lhs, std::array< void const *, N > &&rhs)`
- `template<std::size_t N, typename Arg >`  
`bool Catch::Matchers::Detail::match_all_of (Arg &&, std::array< void const *, N > const &, std::index_sequence<>)`
- `template<typename T, typename... MatcherTs, std::size_t N, typename Arg, std::size_t Idx, std::size_t... Indices>`  
`bool Catch::Matchers::Detail::match_all_of (Arg &&arg, std::array< void const *, N > const &matchers, std::index_sequence< Idx, Indices... >)`
- `template<std::size_t N, typename Arg >`  
`bool Catch::Matchers::Detail::match_any_of (Arg &&, std::array< void const *, N > const &, std::index_sequence<>)`
- `template<typename T, typename... MatcherTs, std::size_t N, typename Arg, std::size_t Idx, std::size_t... Indices>`  
`bool Catch::Matchers::Detail::match_any_of (Arg &&arg, std::array< void const *, N > const &matchers, std::index_sequence< Idx, Indices... >)`
- `std::string Catch::Matchers::Detail::describe_multi_matcher (StringRef combine, std::string const *descriptions_begin, std::string const *descriptions_end)`
- `template<typename... MatcherTs, std::size_t... Idx>`  
`std::string Catch::Matchers::Detail::describe_multi_matcher (StringRef combine, std::array< void const *, sizeof...(MatcherTs)> const &matchers, std::index_sequence< Idx... >)`
- `template<typename MatcherLHS, typename MatcherRHS >`  
`std::enable_if_t< Detail::are_generic_matchers< MatcherLHS, MatcherRHS >::value, Detail::MatchAllOfGeneric< MatcherLHS, MatcherRHS > > Catch::Matchers::operator&& (MatcherLHS const &lhs, MatcherRHS const &rhs)`
- `template<typename MatcherLHS, typename MatcherRHS >`  
`std::enable_if_t< Detail::are_generic_matchers< MatcherLHS, MatcherRHS >::value, Detail::MatchAnyOfGeneric< MatcherLHS, MatcherRHS > > Catch::Matchers::operator|| (MatcherLHS const &lhs, MatcherRHS const &rhs)`
- `template<typename MatcherT >`  
`std::enable_if_t< Detail::is_generic_matcher< MatcherT >::value, Detail::MatchNotOfGeneric< MatcherT > > Catch::Matchers::operator! (MatcherT const &matcher)`  
*Wrap provided generic matcher in generic negator.*
- `template<typename MatcherLHS, typename ArgRHS >`  
`std::enable_if_t< Detail::is_generic_matcher< MatcherLHS >::value, Detail::MatchAllOfGeneric< MatcherLHS, MatcherBase< ArgRHS > > > Catch::Matchers::operator&& (MatcherLHS const &lhs, MatcherBase< ArgRHS > const &rhs)`
- `template<typename ArgLHS, typename MatcherRHS >`  
`std::enable_if_t< Detail::is_generic_matcher< MatcherRHS >::value, Detail::MatchAllOfGeneric< MatcherBase< ArgLHS >, MatcherRHS > > Catch::Matchers::operator&& (MatcherBase< ArgLHS > const &lhs, MatcherRHS const &rhs)`
- `template<typename MatcherLHS, typename ArgRHS >`  
`std::enable_if_t< Detail::is_generic_matcher< MatcherLHS >::value, Detail::MatchAnyOfGeneric< MatcherLHS, MatcherBase< ArgRHS > > > Catch::Matchers::operator|| (MatcherLHS const &lhs, MatcherBase< ArgRHS > const &rhs)`



- `template<typename ArgLHS, typename MatcherRHS >`  
`std::enable_if_t< Detail::is_generic_matcher< MatcherRHS >::value, Detail::MatchAnyOfGeneric<`  
`MatcherBase< ArgLHS >, MatcherRHS > > Catch::Matchers::operator|| (MatcherBase< ArgLHS >`  
`const &lhs, MatcherRHS const &rhs)`
- `IsEmptyMatcher Catch::Matchers::IsEmpty ()`  
*Creates a matcher that accepts empty ranges/containers.*
- `HasSizeMatcher Catch::Matchers::Sizels (std::size_t sz)`  
*Creates a matcher that accepts ranges/containers with specific size.*
- `template<typename Matcher >`  
`std::enable_if_t< Detail::is_matcher< Matcher >::value, SizeMatchesMatcher< Matcher > > Catch::↵`  
`Matchers::Sizels (Matcher &&m)`
- `template<typename T >`  
`std::enable_if_t<!Detail::is_matcher< T >::value, ContainsElementMatcher< T, std::equal_to<> > >`   
`Catch::Matchers::Contains (T &&elem)`
- `template<typename Matcher >`  
`std::enable_if_t< Detail::is_matcher< Matcher >::value, ContainsMatcherMatcher< Matcher > >`   
`Catch::Matchers::Contains (Matcher &&matcher)`  
*Creates a matcher that checks whether a range contains element matching a matcher.*
- `template<typename T, typename Equality >`  
`ContainsElementMatcher< T, Equality > Catch::Matchers::Contains (T &&elem, Equality &&eq)`
- `ExceptionMessageMatcher Catch::Matchers::Message (std::string const &message)`  
*Creates a matcher that checks whether a std derived exception has the provided message.*
- `WithinUlpMatcher Catch::Matchers::WithinULP (double target, uint64_t maxUlpDiff)`  
*Creates a matcher that accepts doubles within certain ULP range of target.*
- `WithinUlpMatcher Catch::Matchers::WithinULP (float target, uint64_t maxUlpDiff)`  
*Creates a matcher that accepts floats within certain ULP range of target.*
- `WithinAbsMatcher Catch::Matchers::WithinAbs (double target, double margin)`  
*Creates a matcher that accepts numbers within certain range of target.*
- `WithinRelMatcher Catch::Matchers::WithinRel (double target, double eps)`  
*Creates a matcher that accepts doubles within certain relative range of target.*
- `WithinRelMatcher Catch::Matchers::WithinRel (double target)`  
*Creates a matcher that accepts doubles within 100\*DBL\_EPS relative range of target.*
- `WithinRelMatcher Catch::Matchers::WithinRel (float target, float eps)`  
*Creates a matcher that accepts doubles within certain relative range of target.*
- `WithinRelMatcher Catch::Matchers::WithinRel (float target)`  
*Creates a matcher that accepts floats within 100\*FLT\_EPS relative range of target.*
- `std::string Catch::Matchers::Detail::finalizeDescription (const std::string &desc)`
- `template<typename T, typename Pred >`  
`PredicateMatcher< T, Pred > Catch::Matchers::Predicate (Pred &&predicate, std::string const &descrip-`  
`tion="")`
- `template<typename Matcher >`  
`AllMatchMatcher< Matcher > Catch::Matchers::AllMatch (Matcher &&matcher)`
- `template<typename Matcher >`  
`NoneMatchMatcher< Matcher > Catch::Matchers::NoneMatch (Matcher &&matcher)`
- `template<typename Matcher >`  
`AnyMatchMatcher< Matcher > Catch::Matchers::AnyMatch (Matcher &&matcher)`
- `StringEqualsMatcher Catch::Matchers::Equals (std::string const &str, CaseSensitive caseSensitivity=Case↵`  
`Sensitive::Yes)`  
*Creates matcher that accepts strings that are exactly equal to str*
- `StringContainsMatcher Catch::Matchers::ContainsSubstring (std::string const &str, CaseSensitive case↵`  
`Sensitivity=CaseSensitive::Yes)`  
*Creates matcher that accepts strings that contain str*
- `EndsWithMatcher Catch::Matchers::EndsWith (std::string const &str, CaseSensitive caseSensitivity=Case↵`  
`Sensitive::Yes)`



- Creates matcher that accepts strings that end with `str`*
  - `StartsWithMatcher` [Catch::Matchers::StartsWith](#) (std::string const &str, CaseSensitive caseSensitivity=CaseSensitive::Yes)
- Creates matcher that accepts strings that start with `str`*
  - `RegexMatcher` [Catch::Matchers::Matches](#) (std::string const &regex, CaseSensitive caseSensitivity=CaseSensitive::Yes)
- Creates matcher that accepts strings matching `regex`*
  - `template<typename T, typename AllocComp = std::allocator<T>, typename AllocMatch = AllocComp>`  
`ContainsMatcher< T, AllocComp, AllocMatch >` [Catch::Matchers::Contains](#) (std::vector< T, AllocComp > const &comparator)
- Creates a matcher that matches vectors that contain all elements in `comparator`*
  - `template<typename T, typename Alloc = std::allocator<T>>`  
`VectorContainsElementMatcher< T, Alloc >` [Catch::Matchers::VectorContains](#) (T const &comparator)
- Creates a matcher that matches vectors that contain `comparator` as an element.*
  - `template<typename T, typename AllocComp = std::allocator<T>, typename AllocMatch = AllocComp>`  
`EqualsMatcher< T, AllocComp, AllocMatch >` [Catch::Matchers::Equals](#) (std::vector< T, AllocComp > const &comparator)
- Creates a matcher that matches vectors that are exactly equal to `comparator`*
  - `template<typename T, typename AllocComp = std::allocator<T>, typename AllocMatch = AllocComp>`  
`ApproxMatcher< T, AllocComp, AllocMatch >` [Catch::Matchers::Approx](#) (std::vector< T, AllocComp > const &comparator)
- Creates a matcher that matches vectors that `comparator` as an element.*
  - `template<typename T, typename AllocComp = std::allocator<T>, typename AllocMatch = AllocComp>`  
`UnorderedEqualsMatcher< T, AllocComp, AllocMatch >` [Catch::Matchers::UnorderedEquals](#) (std::vector< T, AllocComp > const &target)
- Creates a matcher that matches vectors that is equal to `target` modulo permutation.*
  - `std::string` [Catch::getFormattedDuration](#) (double duration)
  - `bool` [Catch::shouldShowDuration](#) (IConfig const &config, double duration)
- Should the reporter show duration of test given current configuration?*
  - `std::string` [Catch::serializeFilters](#) (std::vector< std::string > const &filters)
  - `void` [Catch::defaultListReporters](#) (std::ostream &out, std::vector< ReporterDescription > const &descriptions, Verbosity verbosity)
  - `void` [Catch::defaultListListeners](#) (std::ostream &out, std::vector< ListenerDescription > const &descriptions)
  - `void` [Catch::defaultListTags](#) (std::ostream &out, std::vector< TagInfo > const &tags, bool isFiltered)
  - `void` [Catch::defaultListTests](#) (std::ostream &out, ColourImpl \*streamColour, std::vector< TestCaseHandle > const &tests, bool isFiltered, Verbosity verbosity)
  - `void` [Catch::Detail::registerReporterImpl](#) (std::string const &name, IReporterFactoryPtr reporterPtr)

## Variables

- `const auto` [Catch::Benchmark::Detail::warmup\\_iterations](#) = 10000
- `const auto` [Catch::Benchmark::Detail::warmup\\_time](#) = std::chrono::milliseconds(100)
- `const auto` [Catch::Benchmark::Detail::minimum\\_ticks](#) = 1000
- `const auto` [Catch::Benchmark::Detail::warmup\\_seed](#) = 10000
- `const auto` [Catch::Benchmark::Detail::clock\\_resolution\\_estimation\\_time](#) = std::chrono::milliseconds(500)
- `const auto` [Catch::Benchmark::Detail::clock\\_cost\\_estimation\\_time\\_limit](#) = std::chrono::seconds(1)
- `const auto` [Catch::Benchmark::Detail::clock\\_cost\\_estimation\\_tick\\_limit](#) = 100000
- `const auto` [Catch::Benchmark::Detail::clock\\_cost\\_estimation\\_time](#) = std::chrono::milliseconds(10)
- `const auto` [Catch::Benchmark::Detail::clock\\_cost\\_estimation\\_iterations](#) = 10000
- `constexpr StringRef` [Catch::Detail::unprintableString](#) = "{?}"\_sr
- `constexpr accept_many_t` [Catch::Clara::accept\\_many](#) {}

### 6.1.1 Detailed Description

This is a convenience header for Catch2. It includes **all** of Catch2 headers.

Generally the Catch2 users should use specific includes they need, but this header can be used instead for ease-of-experimentation, or just plain convenience, at the cost of (significantly) increased compilation times.

When a new header is added to either the top level folder, or to the corresponding internal subfolder, it should be added here. Headers added to the various subparts (e.g. matchers, generators, etc...), should go their respective catch-all headers.

This is a convenience header for Catch2's benchmarking. It includes **all** of Catch2 headers related to benchmarking.

Generally the Catch2 users should use specific includes they need, but this header can be used instead for ease-of-experimentation, or just plain convenience, at the cost of (significantly) increased compilation times.

When a new header is added to either the `benchmark` folder, or to the corresponding internal (detail) subfolder, it should be added here.

Wrapper for the CONFIG configuration option

When generating internal unique names, there are two options. Either we mix in the current line number, or mix in an incrementing number. We prefer the latter, using `__COUNTER__`, but users might want to use the former.

Wrapper for the WCHAR configuration option

We want to support platforms that do not provide `wchar_t`, so we sometimes have to disable providing `wchar_t` overloads through Catch2, e.g. the `StringMaker` specialization for `std::wstring`.

This is a convenience header for Catch2's Generator support. It includes **all** of Catch2 headers related to generators.

Generally the Catch2 users should use specific includes they need, but this header can be used instead for ease-of-experimentation, or just plain convenience, at the cost of (significantly) increased compilation times.

When a new header is added to either the `generators` folder, or to the corresponding internal subfolder, it should be added here.

This is a convenience header for Catch2's interfaces. It includes **all** of Catch2 headers related to interfaces.

Generally the Catch2 users should use specific includes they need, but this header can be used instead for ease-of-experimentation, or just plain convenience, at the cost of somewhat increased compilation times.

When a new header is added to either the `interfaces` folder, or to the corresponding internal subfolder, it should be added here.

Wrapper for ANDROID\_LOGWRITE configuration option

We want to default to enabling it when compiled for android, but users of the library should also be able to disable it if they want to.

Wrapper for UNCAUGHT\_EXCEPTIONS configuration option

For some functionality, Catch2 requires to know whether there is an active exception. Because `std::uncaught_exception` is deprecated in C++17, we want to use `std::uncaught_exceptions` if possible.

This is a convenience header for Catch2's Matcher support. It includes **all** of Catch2 headers related to matchers.

Generally the Catch2 users should use specific includes they need, but this header can be used instead for ease-of-experimentation, or just plain convenience, at the cost of increased compilation times.

When a new header is added to either the `matchers` folder, or to the corresponding internal subfolder, it should be added here.

This is a convenience header for Catch2's Reporter support. It includes **all** of Catch2 headers related to reporters, including all reporters.

Generally the Catch2 users should use specific includes they need, but this header can be used instead for ease-of-experimentation, or just plain convenience, at the cost of (significantly) increased compilation times.

When a new header (reporter) is added to either the `reporter` folder, or to the corresponding internal subfolder, it should be added here.

## 6.1.2 Macro Definition Documentation

### 6.1.2.1 CATCH\_INTERNAL\_DEFINE\_EXPRESSION\_OPERATOR

```
#define CATCH_INTERNAL_DEFINE_EXPRESSION_OPERATOR(
    op )
```

**Value:**

```
template<typename RhsT, std::enable_if_t<!std::is_arithmetic<std::remove_reference_t<RhsT>::value,
int> = 0> \
    friend auto operator op ( ExprLhs && lhs, RhsT && rhs ) -> BinaryExpr<LhsT, RhsT const&> { \
        return { static_cast<bool>(lhs.m_lhs op rhs), lhs.m_lhs, #op##_sr, rhs }; \
    } \
template<typename RhsT, std::enable_if_t<std::is_arithmetic<RhsT>::value, int> = 0> \
    friend auto operator op ( ExprLhs && lhs, RhsT rhs ) -> BinaryExpr<LhsT, RhsT> { \
        return { static_cast<bool>(lhs.m_lhs op rhs), lhs.m_lhs, #op##_sr, rhs }; \
    }
```

### 6.1.2.2 CATCH\_REGISTER\_LISTENER

```
#define CATCH_REGISTER_LISTENER(
    listenerType )
```

**Value:**

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS
namespace {
    Catch::ListenerRegistrar<listenerType> INTERNAL_CATCH_UNIQUE_NAME(
        catch_internal_RegistrarFor )( #listenerType );
}
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION
```

### 6.1.2.3 CATCH\_REGISTER\_REPORTER

```
#define CATCH_REGISTER_REPORTER(
    name,
    reporterType )
```

**Value:**

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS
namespace {
    Catch::ReporterRegistrar<reporterType> INTERNAL_CATCH_UNIQUE_NAME(
        catch_internal_RegistrarFor )( name );
}
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION
```

#### 6.1.2.4 CATCH\_REGISTER\_TAG\_ALIAS

```
#define CATCH_REGISTER_TAG_ALIAS(
    alias,
    spec )
```

##### Value:

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
namespace{ Catch::RegistrarForTagAliases INTERNAL_CATCH_UNIQUE_NAME( AutoRegisterTagAlias )( alias,
    spec, CATCH_INTERNAL_LINEINFO ); } \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION
```

#### 6.1.2.5 GENERATE

```
#define GENERATE(
    ... )
```

##### Value:

```
Catch::Generators::generate( INTERNAL_CATCH_STRINGIZE(INTERNAL_CATCH_UNIQUE_NAME(generator)), \
    CATCH_INTERNAL_LINEINFO, \
    [ ]{ using namespace Catch::Generators; return makeGenerators( __VA_ARGS__
); } )
```

#### 6.1.2.6 GENERATE\_COPY

```
#define GENERATE_COPY(
    ... )
```

##### Value:

```
Catch::Generators::generate( INTERNAL_CATCH_STRINGIZE(INTERNAL_CATCH_UNIQUE_NAME(generator)), \
    CATCH_INTERNAL_LINEINFO, \
    [=]{ using namespace Catch::Generators; return makeGenerators( __VA_ARGS__
); } )
```

#### 6.1.2.7 GENERATE\_REF

```
#define GENERATE_REF(
    ... )
```

##### Value:

```
Catch::Generators::generate( INTERNAL_CATCH_STRINGIZE(INTERNAL_CATCH_UNIQUE_NAME(generator)), \
    CATCH_INTERNAL_LINEINFO, \
    [&]{ using namespace Catch::Generators; return makeGenerators( __VA_ARGS__
); } )
```

### 6.1.2.8 INTERNAL\_CATCH\_BENCHMARK

```
#define INTERNAL_CATCH_BENCHMARK(  
    BenchmarkName,  
    name,  
    benchmarkIndex )
```

**Value:**

```
if( Catch::Benchmark::Benchmark BenchmarkName{name} ) \  
    BenchmarkName = [&](int benchmarkIndex)
```

### 6.1.2.9 INTERNAL\_CATCH\_BENCHMARK\_ADVANCED

```
#define INTERNAL_CATCH_BENCHMARK_ADVANCED(  
    BenchmarkName,  
    name )
```

**Value:**

```
if( Catch::Benchmark::Benchmark BenchmarkName{name} ) \  
    BenchmarkName = [&]
```

### 6.1.2.10 INTERNAL\_CATCH\_CAPTURE

```
#define INTERNAL_CATCH_CAPTURE(  
    varName,  
    macroName,  
    ... )
```

**Value:**

```
Catch::Capturer varName( macroName, CATCH_INTERNAL_LINEINFO, Catch::ResultWas::Info, #__VA_ARGS__ ); \  
varName.captureValues( 0, __VA_ARGS__ )
```

### 6.1.2.11 INTERNAL\_CATCH\_DECLARE\_SIG\_TEST1

```
#define INTERNAL_CATCH_DECLARE_SIG_TEST1(  
    TestName,  
    signature )
```

**Value:**

```
template<INTERNAL_CATCH_REMOVE_PARENS(signature)>\  
static void TestName()
```

### 6.1.2.12 INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_METHOD1

```
#define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD1(
    TestName,
    ClassName,
    signature )
```

**Value:**

```
template<typename TestType> \
struct TestName : INTERNAL_CATCH_REMOVE_PARENS(ClassName)<TestType> { \
    void test();\
}
```

### 6.1.2.13 INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_METHOD\_X

```
#define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD_X(
    TestName,
    ClassName,
    signature,
    ... )
```

**Value:**

```
template<INTERNAL_CATCH_REMOVE_PARENS(signature)> \
struct TestName : INTERNAL_CATCH_REMOVE_PARENS(ClassName)<__VA_ARGS__> { \
    void test();\
}
```

### 6.1.2.14 INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_X

```
#define INTERNAL_CATCH_DECLARE_SIG_TEST_X(
    TestName,
    signature,
    ... )
```

**Value:**

```
template<INTERNAL_CATCH_REMOVE_PARENS(signature)>\
static void TestName()
```

### 6.1.2.15 INTERNAL\_CATCH\_DEFINE\_SIG\_TEST1

```
#define INTERNAL_CATCH_DEFINE_SIG_TEST1(
    TestName,
    signature )
```

**Value:**

```
template<INTERNAL_CATCH_REMOVE_PARENS(signature)>\
static void TestName()
```

### 6.1.2.16 INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_METHOD1

```
#define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD1(
    TestName,
    signature )
```

**Value:**

```
template<typename TestType> \
void INTERNAL_CATCH_MAKE_NAMESPACE( TestName )::TestName<TestType>::test()
```

### 6.1.2.17 INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_METHOD\_X

```
#define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD_X(
    TestName,
    signature,
    ... )
```

**Value:**

```
template<INTERNAL_CATCH_REMOVE_PARENS( signature )> \
void INTERNAL_CATCH_MAKE_NAMESPACE( TestName )::TestName<__VA_ARGS__>::test()
```

### 6.1.2.18 INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_X

```
#define INTERNAL_CATCH_DEFINE_SIG_TEST_X(
    TestName,
    signature,
    ... )
```

**Value:**

```
template<INTERNAL_CATCH_REMOVE_PARENS( signature )>\
static void TestName()
```

### 6.1.2.19 INTERNAL\_CATCH\_DYNAMIC\_SECTION

```
#define INTERNAL_CATCH_DYNAMIC_SECTION(
    ... )
```

**Value:**

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_UNUSED_VARIABLE_WARNINGS \
if ( Catch::Section const& INTERNAL_CATCH_UNIQUE_NAME( catch_internal_Section ) = Catch::SectionInfo(
    CATCH_INTERNAL_LINEINFO, (Catch::ReusableStringStream() << __VA_ARGS__).str() ) ) \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION
```

### 6.1.2.20 INTERNAL\_CATCH\_ELSE

```
#define INTERNAL_CATCH_ELSE(
    macroName,
    resultDisposition,
    ... )
```

#### Value:

```
INTERNAL_CATCH_TEST( macroName, resultDisposition, __VA_ARGS__ ); \
if( !Catch::getResultCapture().lastAssertionPassed() )
```

### 6.1.2.21 INTERNAL\_CATCH\_IF

```
#define INTERNAL_CATCH_IF(
    macroName,
    resultDisposition,
    ... )
```

#### Value:

```
INTERNAL_CATCH_TEST( macroName, resultDisposition, __VA_ARGS__ ); \
if( Catch::getResultCapture().lastAssertionPassed() )
```

### 6.1.2.22 INTERNAL\_CATCH\_METHOD\_AS\_TEST\_CASE

```
#define INTERNAL_CATCH_METHOD_AS_TEST_CASE(
    QualifiedMethod,
    ... )
```

#### Value:

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
namespace{ Catch::AutoReg INTERNAL_CATCH_UNIQUE_NAME( autoRegistrar )( Catch::makeTestInvoker(
&QualifiedMethod ), CATCH_INTERNAL_LINEINFO, "&" #QualifiedMethod, Catch::NameAndTags{ __VA_ARGS__ }
); } /* NOLINT */ \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION
```

### 6.1.2.23 INTERNAL\_CATCH\_MSG

```
#define INTERNAL_CATCH_MSG(
    macroName,
    messageType,
    resultDisposition,
    ... )
```

#### Value:

```
do { \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    Catch::StringRef(), resultDisposition ); \
    catchAssertionHandler.handleMessage( messageType, ( Catch::MessageStream() « __VA_ARGS__ +
    ::Catch::StreamEndStop() ).m_stream.str() ); \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( false )
```



## 6.1.2.24 INTERNAL\_CATCH\_NO\_THROW

```
#define INTERNAL_CATCH_NO_THROW(
    macroName,
    resultDisposition,
    ... )
```

## Value:

```
do { \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    CATCH_INTERNAL_STRINGIFY(__VA_ARGS__), resultDisposition ); \
    try { \
        static_cast<void>(__VA_ARGS__); \
        catchAssertionHandler.handleExceptionNotThrownAsExpected(); \
    } \
    catch( ... ) { \
        catchAssertionHandler.handleUnexpectedInflightException(); \
    } \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( false )
```

## 6.1.2.25 INTERNAL\_CATCH\_NTTP\_1

```
#define INTERNAL_CATCH_NTTP_1(
    signature,
    ... )
```

## Value:

```
template<INTERNAL_CATCH_REMOVE_PARENS(signature)> struct Nttp{};\
template<INTERNAL_CATCH_REMOVE_PARENS(signature)>\
constexpr auto get_wrapper() noexcept -> Nttp<__VA_ARGS__> { return {}; } \
template<template<INTERNAL_CATCH_REMOVE_PARENS(signature)> class...> struct NttpTemplateTypeList{};\
template<template<INTERNAL_CATCH_REMOVE_PARENS(signature)> class...Cs>\
constexpr auto get_wrapper() noexcept -> NttpTemplateTypeList<Cs...> { return {}; } \
\
template< template<INTERNAL_CATCH_REMOVE_PARENS(signature)> class Container,\
template<INTERNAL_CATCH_REMOVE_PARENS(signature)> class List,\
INTERNAL_CATCH_REMOVE_PARENS(signature)>\
struct rewrap<NttpTemplateTypeList<Container>, List<__VA_ARGS__> { using type =
TypeList<Container<__VA_ARGS__>; };\
template< template<INTERNAL_CATCH_REMOVE_PARENS(signature)> class Container,\
template<INTERNAL_CATCH_REMOVE_PARENS(signature)> class List,\
INTERNAL_CATCH_REMOVE_PARENS(signature), typename...Elements>\
struct rewrap<NttpTemplateTypeList<Container>, List<__VA_ARGS__>, Elements...> { using type = typename
append<TypeList<Container<__VA_ARGS__>, typename rewrap<NttpTemplateTypeList<Container>,
Elements...>::type>::type; };\
template<template <typename...> class Final, template<INTERNAL_CATCH_REMOVE_PARENS(signature)>
class...Containers, typename...Types>\
struct create<Final, NttpTemplateTypeList<Containers...>, TypeList<Types...> { using type = typename
append<Final<>, typename rewrap<NttpTemplateTypeList<Containers>, Types...>::type...>::type; };
```

## 6.1.2.26 INTERNAL\_CATCH\_NTTP\_REGISTER

```
#define INTERNAL_CATCH_NTTP_REGISTER(
    TestFunc,
    signature,
    ... )
```

## Value:

```
template<INTERNAL_CATCH_REMOVE_PARENS(signature)>\
void reg_test(Nttp<__VA_ARGS__>, Catch::NameAndTags nameAndTags)\
{\
    Catch::AutoReg( Catch::makeTestInvoker(&TestFunc<__VA_ARGS__>), CATCH_INTERNAL_LINEINFO,
    Catch::StringRef(), nameAndTags);\
}
```

### 6.1.2.27 INTERNAL\_CATCH\_NTTP\_REGISTER0

```
#define INTERNAL_CATCH_NTTP_REGISTER0(
    TestFunc,
    signature )
```

#### Value:

```
template<typename Type>\
void reg_test(TypeList<Type>, Catch::NameAndTags nameAndTags)\
{\
    Catch::AutoReg( Catch::makeTestInvoker(&TestFunc<Type>), CATCH_INTERNAL_LINEINFO,
    Catch::StringRef(), nameAndTags);\
}
```

### 6.1.2.28 INTERNAL\_CATCH\_NTTP\_REGISTER\_METHOD

```
#define INTERNAL_CATCH_NTTP_REGISTER_METHOD(
    TestName,
    signature,
    ... )
```

#### Value:

```
template<INTERNAL_CATCH_REMOVE_PARENS(signature)>\
void reg_test(Nttp<__VA_ARGS__>, Catch::StringRef className, Catch::NameAndTags nameAndTags)\
{\
    Catch::AutoReg( Catch::makeTestInvoker(&TestName<__VA_ARGS__>::test), CATCH_INTERNAL_LINEINFO,
    className, nameAndTags);\
}
```

### 6.1.2.29 INTERNAL\_CATCH\_NTTP\_REGISTER\_METHOD0

```
#define INTERNAL_CATCH_NTTP_REGISTER_METHOD0(
    TestName,
    signature,
    ... )
```

#### Value:

```
template<typename Type>\
void reg_test(TypeList<Type>, Catch::StringRef className, Catch::NameAndTags nameAndTags)\
{\
    Catch::AutoReg( Catch::makeTestInvoker(&TestName<Type>::test), CATCH_INTERNAL_LINEINFO, className,
    nameAndTags);\
}
```

### 6.1.2.30 INTERNAL\_CATCH\_REGISTER\_ENUM

```
#define INTERNAL_CATCH_REGISTER_ENUM(
    enumName,
    ... )
```

#### Value:

```
namespace Catch { \
    template<> struct StringMaker<enumName> { \
        static std::string convert( enumName value ) { \
            static const auto& enumInfo =
            ::Catch::getMutableRegistryHub().getMutableEnumValuesRegistry().registerEnum( #enumName,
            #__VA_ARGS__, { __VA_ARGS__ } ); \
            return static_cast<std::string>(enumInfo.lookup( static_cast<int>( value ) )); \
        } \
    }; \
}
```

## 6.1.2.31 INTERNAL\_CATCH\_REGISTER\_TESTCASE

```
#define INTERNAL_CATCH_REGISTER_TESTCASE(
    Function,
    ... )
```

**Value:**

```
do { \
    CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
    CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
    Catch::AutoReg INTERNAL_CATCH_UNIQUE_NAME( autoRegistrar )( Catch::makeTestInvoker( Function ),
    CATCH_INTERNAL_LINEINFO, Catch::StringRef(), Catch::NameAndTags{ __VA_ARGS__ } ); /* NOLINT */ \
    CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
} while(false)
```

## 6.1.2.32 INTERNAL\_CATCH\_SECTION

```
#define INTERNAL_CATCH_SECTION(
    ... )
```

**Value:**

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_UNUSED_VARIABLE_WARNINGS \
if( Catch::Section const& INTERNAL_CATCH_UNIQUE_NAME( catch_internal_section ) = Catch::SectionInfo(
    CATCH_INTERNAL_LINEINFO, __VA_ARGS__ ) ) \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION
```

## 6.1.2.33 INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_2

```
#define INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_2(
    TestName,
    TestFunc,
    Name,
    Tags,
    TmplList )
```

**Value:**

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_VARIABLE_WARNINGS \
template<typename TestType> static void TestFunc(); \
namespace { \
    namespace INTERNAL_CATCH_MAKE_NAMESPACE( TestName ) { \
        INTERNAL_CATCH_TYPE_GEN \
        template<typename... Types> \
        struct TestName { \
            void reg_tests() { \
                size_t index = 0; \
                using expander = size_t[]; \
                (void)expander{ Catch::AutoReg( Catch::makeTestInvoker( &TestFunc<Types> ), \
                CATCH_INTERNAL_LINEINFO, Catch::StringRef(), Catch::NameAndTags{ Name " - " + \
                std::string(INTERNAL_CATCH_STRINGIZE(TmplList)) + " - " + std::to_string(index), Tags } ), \
                index++)... }; /* NOLINT */ \
            } \
        }; \
        static int INTERNAL_CATCH_UNIQUE_NAME( globalRegistrar ) = []() { \
            using TestInit = typename convert<TestName, TmplList>::type; \
            TestInit t; \
            t.reg_tests(); \
            return 0; \
        }(); \
    } \
    CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
    template<typename TestType> \
    static void TestFunc()
```

### 6.1.2.34 INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_METHOD\_2

```
#define INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD_2(
    TestNameClass,
    TestName,
    ClassName,
    Name,
    Tags,
    TmplList )
```

#### Value:

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_VARIABLE_WARNINGS \
template<typename TestType> \
struct TestName : INTERNAL_CATCH_REMOVE_PARENS(ClassName <TestType>) { \
    void test(); \
}; \
namespace { \
namespace INTERNAL_CATCH_MAKE_NAMESPACE(TestName) { \
    INTERNAL_CATCH_TYPE_GEN \
    template<typename...Types> \
    struct TestNameClass{ \
        void reg_tests(){ \
            size_t index = 0; \
            using expander = size_t[]; \
            (void)expander{(Catch::AutoReg( Catch::makeTestInvoker( &TestName<Types>::test ), \
CATCH_INTERNAL_LINEINFO, #ClassName, Catch::NameAndTags{ Name " - " + \
std::string(INTERNAL_CATCH_STRINGIZE(TmplList)) + " - " + std::to_string(index), Tags } ), \
index++)... }; /* NOLINT */ \
        } \
    }; \
    static int INTERNAL_CATCH_UNIQUE_NAME( globalRegistrar ) = [](){ \
        using TestInit = typename convert<TestNameClass, TmplList>::type; \
        TestInit t; \
        t.reg_tests(); \
        return 0; \
    }(); \
} \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
template<typename TestType> \
void TestName<TestType>::test()
```

### 6.1.2.35 INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_2

```
#define INTERNAL_CATCH_TEMPLATE_TEST_CASE_2(
    TestName,
    TestFunc,
    Name,
    Tags,
    Signature,
    ... )
```

#### Value:

```
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
CATCH_INTERNAL_SUPPRESS_ZERO_VARIADIC_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_VARIABLE_WARNINGS \
INTERNAL_CATCH_DECLARE_SIG_TEST(TestFunc, INTERNAL_CATCH_REMOVE_PARENS(Signature)); \
namespace { \
namespace INTERNAL_CATCH_MAKE_NAMESPACE(TestName) { \
    INTERNAL_CATCH_TYPE_GEN \
    INTERNAL_CATCH_NTTP_GEN(INTERNAL_CATCH_REMOVE_PARENS(Signature)) \
    INTERNAL_CATCH_NTTP_REG_GEN(TestFunc, INTERNAL_CATCH_REMOVE_PARENS(Signature)) \
    template<typename...Types> \
    struct TestName{ \
        TestName() { \
            size_t index = 0; \
        } \
    }; \
}
```

```

        constexpr char const* tmpl_types[] =
{CATCH_REC_LIST(INTERNAL_CATCH_STRINGIZE_WITHOUT_PARENS, __VA_ARGS__)};\
    using expander = size_t[];\
    (void)expander{(reg_test(Types{}, Catch::NameAndTags{ Name " - " +
std::string(tmpl_types[index]), Tags } ), index++)... };/* NOLINT */ \
    }\
};\
    static int INTERNAL_CATCH_UNIQUE_NAME( globalRegistrar ) = [](){\
    TestName<INTERNAL_CATCH_MAKE_TYPE_LISTS_FROM_TYPES(__VA_ARGS__)>();\
    return 0;\
}();\
}\
}\
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
INTERNAL_CATCH_DEFINE_SIG_TEST (TestFunc, INTERNAL_CATCH_REMOVE_PARENS (Signature))

```

### 6.1.2.36 INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_2

```
#define INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD_2(
```

```

    TestNameClass,
    TestName,
    ClassName,
    Name,
    Tags,
    Signature,
    ... )

```

#### Value:

```

CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
CATCH_INTERNAL_SUPPRESS_ZERO_VARIADIC_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS \
CATCH_INTERNAL_SUPPRESS_UNUSED_VARIABLE_WARNINGS \
namespace {\
namespace INTERNAL_CATCH_MAKE_NAMESPACE(TestName){ \
    INTERNAL_CATCH_TYPE_GEN\
    INTERNAL_CATCH_NTTP_GEN(INTERNAL_CATCH_REMOVE_PARENS(Signature))\
    INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD(TestName, ClassName,\
INTERNAL_CATCH_REMOVE_PARENS(Signature));\
    INTERNAL_CATCH_NTTP_REG_METHOD_GEN(TestName, INTERNAL_CATCH_REMOVE_PARENS(Signature))\
    template<typename...Types> \
    struct TestNameClass{\
        TestNameClass(){\
            size_t index = 0;\
            constexpr char const* tmpl_types[] = \
{CATCH_REC_LIST(INTERNAL_CATCH_STRINGIZE_WITHOUT_PARENS, __VA_ARGS__)};\
            using expander = size_t[];\
            (void)expander{(reg_test(Types{}, #ClassName, Catch::NameAndTags{ Name " - " +
std::string(tmpl_types[index]), Tags } ), index++)... };/* NOLINT */ \
        }\
    }; \
    static int INTERNAL_CATCH_UNIQUE_NAME( globalRegistrar ) = [](){\
        TestNameClass<INTERNAL_CATCH_MAKE_TYPE_LISTS_FROM_TYPES(__VA_ARGS__)>();\
        return 0;\
    }();\
}\
}\
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD(TestName, INTERNAL_CATCH_REMOVE_PARENS(Signature))

```

### 6.1.2.37 INTERNAL\_CATCH\_TEST

```
#define INTERNAL_CATCH_TEST(
    macroName,
    resultDisposition,
    ... )

```

**Value:**

```

do { /* NOLINT(bugprone-infinite-loop) */ \
    /* The expression should not be evaluated, but warnings should hopefully be checked */ \
    CATCH_INTERNAL_IGNORE_BUT_WARN(__VA_ARGS__); \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    CATCH_INTERNAL_STRINGIFY(__VA_ARGS__), resultDisposition ); \
    INTERNAL_CATCH_TRY { \
        CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
        CATCH_INTERNAL_SUPPRESS_PARENTHESES_WARNINGS \
        catchAssertionHandler.handleExpr( Catch::Decomposer() <= __VA_ARGS__ ); \
        CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
    } INTERNAL_CATCH_CATCH( catchAssertionHandler ) \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( (void)0, (false) && static_cast<const bool&>( !( __VA_ARGS__ ) ) )

```

**6.1.2.38 INTERNAL\_CATCH\_TEST\_CASE\_METHOD2**

```

#define INTERNAL_CATCH_TEST_CASE_METHOD2(
    TestName,
    ClassName,
    ... )

```

**Value:**

```

CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
namespace{ \
    struct TestName : INTERNAL_CATCH_REMOVE_PARENS(ClassName) { \
        void test(); \
    }; \
    Catch::AutoReg INTERNAL_CATCH_UNIQUE_NAME( autoRegistrar ) ( Catch::makeTestInvoker(
    &TestName::test ), CATCH_INTERNAL_LINEINFO, #ClassName, Catch::NameAndTags{ __VA_ARGS__ } ); /*
    NOLINT */ \
} \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
void TestName::test()

```

**6.1.2.39 INTERNAL\_CATCH\_TESTCASE2**

```

#define INTERNAL_CATCH_TESTCASE2(
    TestName,
    ... )

```

**Value:**

```

static void TestName(); \
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
namespace{ \
    Catch::AutoReg INTERNAL_CATCH_UNIQUE_NAME( autoRegistrar ) ( Catch::makeTestInvoker(
    &TestName ), CATCH_INTERNAL_LINEINFO, Catch::StringRef(), Catch::NameAndTags{ __VA_ARGS__ } ); } /*
    NOLINT */ \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
static void TestName()

```

## 6.1.2.40 INTERNAL\_CATCH\_THROWS

```
#define INTERNAL_CATCH_THROWS(
    macroName,
    resultDisposition,
    ... )
```

**Value:**

```
do { \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    CATCH_INTERNAL_STRINGIFY(__VA_ARGS__), resultDisposition); \
    if( catchAssertionHandler.allowThrows() ) \
        try { \
            static_cast<void>(__VA_ARGS__); \
            catchAssertionHandler.handleUnexpectedExceptionNotThrown(); \
        } \
        catch( ... ) { \
            catchAssertionHandler.handleExceptionThrownAsExpected(); \
        } \
    else \
        catchAssertionHandler.handleThrowingCallSkipped(); \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( false )
```

## 6.1.2.41 INTERNAL\_CATCH\_THROWS\_AS

```
#define INTERNAL_CATCH_THROWS_AS(
    macroName,
    exceptionType,
    resultDisposition,
    expr )
```

**Value:**

```
do { \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    CATCH_INTERNAL_STRINGIFY(expr) ", " CATCH_INTERNAL_STRINGIFY(exceptionType), resultDisposition ); \
    if( catchAssertionHandler.allowThrows() ) \
        try { \
            static_cast<void>(expr); \
            catchAssertionHandler.handleUnexpectedExceptionNotThrown(); \
        } \
        catch( exceptionType const& ) { \
            catchAssertionHandler.handleExceptionThrownAsExpected(); \
        } \
        catch( ... ) { \
            catchAssertionHandler.handleUnexpectedInflightException(); \
        } \
    else \
        catchAssertionHandler.handleThrowingCallSkipped(); \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( false )
```

## 6.1.2.42 INTERNAL\_CATCH\_THROWS\_MATCHES

```
#define INTERNAL_CATCH_THROWS_MATCHES(
    macroName,
    exceptionType,
    resultDisposition,
    matcher,
    ... )
```

**Value:**

```

do { \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    CATCH_INTERNAL_STRINGIFY(__VA_ARGS__) ", " CATCH_INTERNAL_STRINGIFY(exceptionType) ", "
    CATCH_INTERNAL_STRINGIFY(matcher), resultDisposition ); \
    if( catchAssertionHandler.allowThrows() ) \
        try { \
            static_cast<void>(__VA_ARGS__ ); \
            catchAssertionHandler.handleUnexpectedExceptionNotThrown(); \
        } \
        catch( exceptionType const& ex ) { \
            catchAssertionHandler.handleExpr( Catch::makeMatchExpr( ex, matcher, #matcher##_catch_sr )
    ); \
        } \
        catch( ... ) { \
            catchAssertionHandler.handleUnexpectedInflightException(); \
        } \
    else \
        catchAssertionHandler.handleThrowingCallSkipped(); \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( false )

```

**6.1.2.43 INTERNAL\_CATCH\_THROWS\_STR\_MATCHES**

```

#define INTERNAL_CATCH_THROWS_STR_MATCHES(
    macroName,
    resultDisposition,
    matcher,
    ... )

```

**Value:**

```

do { \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    CATCH_INTERNAL_STRINGIFY(__VA_ARGS__) ", " CATCH_INTERNAL_STRINGIFY(matcher), resultDisposition ); \
    if( catchAssertionHandler.allowThrows() ) \
        try { \
            static_cast<void>(__VA_ARGS__); \
            catchAssertionHandler.handleUnexpectedExceptionNotThrown(); \
        } \
        catch( ... ) { \
            Catch::handleExceptionMatchExpr( catchAssertionHandler, matcher, #matcher##_catch_sr ); \
        } \
    else \
        catchAssertionHandler.handleThrowingCallSkipped(); \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( false )

```

**6.1.2.44 INTERNAL\_CATCH\_TRANSLATE\_EXCEPTION2**

```

#define INTERNAL_CATCH_TRANSLATE_EXCEPTION2(
    translatorName,
    signature )

```

**Value:**

```

static std::string translatorName( signature ); \
CATCH_INTERNAL_START_WARNINGS_SUPPRESSION \
CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS \
namespace{ Catch::ExceptionTranslatorRegistrar INTERNAL_CATCH_UNIQUE_NAME(
    catch_internal_ExceptionRegistrar )( &translatorName ); } \
CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION \
static std::string translatorName( signature )

```



### 6.1.2.45 INTERNAL\_CHECK\_THAT

```
#define INTERNAL_CHECK_THAT(
    macroName,
    matcher,
    resultDisposition,
    arg )
```

#### Value:

```
do { \
    Catch::AssertionHandler catchAssertionHandler( macroName##_catch_sr, CATCH_INTERNAL_LINEINFO,
    CATCH_INTERNAL_STRINGIFY(arg) ", " CATCH_INTERNAL_STRINGIFY(matcher), resultDisposition ); \
    INTERNAL_CATCH_TRY { \
        catchAssertionHandler.handleExpr( Catch::makeMatchExpr( arg, matcher, #matcher##_catch_sr ) ); \
    } INTERNAL_CATCH_CATCH( catchAssertionHandler ) \
    INTERNAL_CATCH_REACT( catchAssertionHandler ) \
} while( false )
```

## 6.1.3 Enumeration Type Documentation

### 6.1.3.1 ColourMode

```
enum Catch::ColourMode : std::uint8_t [strong]
```

#### Enumerator

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| PlatformDefault | Let Catch2 pick implementation based on platform detection. |
| ANSI            | Use ANSI colour code escapes.                               |
| Win32           | Use Win32 console colour API.                               |
| None            | Don't use any colour.                                       |

### 6.1.3.2 GenerateFrom

```
enum Catch::GenerateFrom [strong]
```

#### Enumerator

|         |                                                                    |
|---------|--------------------------------------------------------------------|
| Default | Currently equivalent to RandomDevice, but can change at any point. |
|---------|--------------------------------------------------------------------|

### 6.1.3.3 ResultType

```
enum Catch::Clara::Detail::ResultType [strong]
```

Denotes type of a parsing result.

## Enumerator

|              |                                                    |
|--------------|----------------------------------------------------|
| Ok           | No errors.                                         |
| LogicError   | Error in user-specified arguments for construction |
| RuntimeError | Error in parsing inputs.                           |

## 6.1.4 Function Documentation

### 6.1.4.1 Contains() [1/2]

```
template<typename T >
std::enable_if_t

```

Creates a matcher that checks whether a range contains a specific element.

Uses `std::equal_to` to do the comparison

### 6.1.4.2 Contains() [2/2]

```
template<typename T , typename Equality >
ContainsElementMatcher<T, Equality> Catch::Matchers::Contains (
    T && elem,
    Equality && eq )
```

Creates a matcher that checks whether a range contains a specific element.

Uses `eq` to do the comparisons

### 6.1.4.3 convertIntoString()

```
std::string Catch::Detail::convertIntoString (
    StringRef string )
```

Encases `string` in quotes, and escapes invisibles if user requested it via CLI

### 6.1.4.4 defaultListListeners()

```
void Catch::defaultListListeners (
    std::ostream & out,
    std::vector< ListenerDescription > const & descriptions )
```

Lists listeners descriptions to the provided stream in user-friendly format

#### 6.1.4.5 defaultListReporters()

```
void Catch::defaultListReporters (
    std::ostream & out,
    std::vector< ReporterDescription > const & descriptions,
    Verbosity verbosity )
```

Lists reporter descriptions to the provided stream in user-friendly format

Used as the default listing implementation by the first party reporter bases. The output should be backwards compatible with the output of Catch2 v2 binaries.

#### 6.1.4.6 defaultListTags()

```
void Catch::defaultListTags (
    std::ostream & out,
    std::vector< TagInfo > const & tags,
    bool isFiltered )
```

Lists tag information to the provided stream in user-friendly format

Used as the default listing implementation by the first party reporter bases. The output should be backwards compatible with the output of Catch2 v2 binaries.

#### 6.1.4.7 defaultListTests()

```
void Catch::defaultListTests (
    std::ostream & out,
    ColourImpl * streamColour,
    std::vector< TestCaseHandle > const & tests,
    bool isFiltered,
    Verbosity verbosity )
```

Lists test case information to the provided stream in user-friendly format

Used as the default listing implementation by the first party reporter bases. The output is backwards compatible with the output of Catch2 v2 binaries, and also supports the format specific to the old `--list-test-names-only` option, for people who used it in integrations.

#### 6.1.4.8 makeStream()

```
auto Catch::makeStream (
    std::string const & filename ) -> Detail::unique_ptr< IStream >
```

Creates a stream wrapper that writes to specific file.

Also recognizes 4 special filenames

- `-` for stdout
- `stdout` for stdout
- `stderr` for stderr
- `debug` for platform specific debugging output

## Exceptions

|           |                                          |
|-----------|------------------------------------------|
| <i>if</i> | passed an unrecognized %-prefixed stream |
|-----------|------------------------------------------|

**6.1.4.9 operator&&() [1/2]**

```
template<typename ArgT >
MatchAllOf<ArgT> Catch::Matchers::Detail::operator&& (
    MatchAllOf< ArgT > const & lhs,
    MatcherBase< ArgT > const & rhs ) [delete]
```

lvalue overload is intentionally deleted, users should not be trying to compose stored composition matchers

**6.1.4.10 operator&&() [2/2]**

```
template<typename ArgT >
MatchAllOf<ArgT> Catch::Matchers::Detail::operator&& (
    MatcherBase< ArgT > const & lhs,
    MatchAllOf< ArgT > const & rhs ) [delete]
```

lvalue overload is intentionally deleted, users should not be trying to compose stored composition matchers

**6.1.4.11 operator"|"() [1/2]**

```
template<typename ArgT >
MatchAnyOf<ArgT> Catch::Matchers::Detail::operator|| (
    MatchAnyOf< ArgT > const & lhs,
    MatcherBase< ArgT > const & rhs ) [delete]
```

lvalue overload is intentionally deleted, users should not be trying to compose stored composition matchers

**6.1.4.12 operator"|"() [2/2]**

```
template<typename ArgT >
MatchAnyOf<ArgT> Catch::Matchers::Detail::operator|| (
    MatcherBase< ArgT > const & lhs,
    MatchAnyOf< ArgT > const & rhs ) [delete]
```

lvalue overload is intentionally deleted, users should not be trying to compose stored composition matchers

#### 6.1.4.13 parseReporterSpec()

```
Optional<ReporterSpec> Catch::parseReporterSpec (
    StringRef reporterSpec )
```

Parses provided reporter spec string into

Returns empty optional on errors, e.g.

- field that is not first and not a key+value pair
- duplicated keys in kv pair
- unknown catch reporter option
- empty key/value in an custom kv pair
- ...

#### 6.1.4.14 Predicate()

```
template<typename T , typename Pred >
PredicateMatcher<T, Pred> Catch::Matchers::Predicate (
    Pred && predicate,
    std::string const & description = "" )
```

Creates a matcher that calls delegates `match` to the provided predicate.

The user has to explicitly specify the argument type to the matcher

#### 6.1.4.15 registerReporterImpl()

```
void Catch::Detail::registerReporterImpl (
    std::string const & name,
    IReporterFactoryPtr reporterPtr )
```

Indirection for reporter registration, so that the error handling is independent on the reporter's concrete type

#### 6.1.4.16 ulpDistance()

```
template<typename FP >
uint64_t Catch::ulpDistance (
    FP lhs,
    FP rhs )
```

Calculates the ULP distance between two floating point numbers

The ULP distance of two floating point numbers is the count of valid floating point numbers representable between them.

There are some exceptions between how this function counts the distance, and the interpretation of the standard as implemented. by e.g. `nextafter`. For this function it always holds that:

- $(x == y) \Rightarrow \text{ulpDistance}(x, y) == 0$  (so  $\text{ulpDistance}(-0, 0) == 0$ )
- $\text{ulpDistance}(\text{maxFinite}, \text{INF}) == 1$
- $\text{ulpDistance}(x, -x) == 2 * \text{ulpDistance}(x, 0)$

#### Precondition

```
!isnan( lhs )
!isnan( rhs )
```

floating point numbers are represented in IEEE-754 format



# Index

Action  
  Action< TSeq >, 22  
Action< TSeq >, 21  
  Action, 22  
add\_global\_action  
  Model< TSeq >, 180  
AdjList, 23  
  AdjList, 23  
  read\_edgelist, 24  
Agent< TSeq >, 24  
ANSI  
  catch\_amalgamated.hpp, 345  
  
Catch::always\_false< T >, 28  
Catch::Approx, 30  
Catch::AssertionHandler, 34  
Catch::AssertionInfo, 34  
Catch::AssertionReaction, 35  
Catch::AssertionResult, 36  
Catch::AssertionResultData, 37  
Catch::AssertionStats, 38  
Catch::AutomakeReporter, 39  
Catch::AutoReg, 40  
Catch::Benchmark::Benchmark, 42  
Catch::Benchmark::Chronometer, 63  
Catch::Benchmark::Detail::BenchmarkFunction, 42  
Catch::Benchmark::Detail::bootstrap\_analysis, 46  
Catch::Benchmark::Detail::ChronometerConcept, 63  
Catch::Benchmark::Detail::ChronometerModel< Clock  
  >, 64  
Catch::Benchmark::Detail::CompleteInvoker< Result >,  
  72  
Catch::Benchmark::Detail::CompleteInvoker< void >,  
  73  
Catch::Benchmark::Detail::CompleteType< T >, 73  
Catch::Benchmark::Detail::CompleteType< void >, 73  
Catch::Benchmark::Detail::CompleteType<       void  
  >::type, 280  
Catch::Benchmark::Detail::is\_related< T, U >, 138  
Catch::Benchmark::Detail::ObjectStorage< T, Destruct  
  >, 189  
Catch::Benchmark::Detail::repeater< Fun >, 209  
Catch::Benchmark::Environment< Clock >, 96  
Catch::Benchmark::EnvironmentEstimate< Duration >,  
  97  
Catch::Benchmark::Estimate< Duration >, 99  
Catch::Benchmark::ExecutionPlan< Duration >, 105  
Catch::Benchmark::now< Clock >, 189  
Catch::Benchmark::OutlierClassification, 192  
Catch::Benchmark::SampleAnalysis< Duration >, 225  
  
Catch::Benchmark::Timing< Duration, Result >, 273  
Catch::BenchmarkInfo, 43  
Catch::BenchmarkStats< Duration >, 44  
Catch::BinaryExpr< LhsT, RhsT >, 45  
Catch::Capturer, 61  
Catch::Clara::accept\_many\_t, 21  
Catch::Clara::Arg, 32  
Catch::Clara::Args, 33  
Catch::Clara::Detail::BasicResult< T >, 41  
Catch::Clara::Detail::BoundFlagLambda< L >, 47  
Catch::Clara::Detail::BoundFlagRef, 49  
Catch::Clara::Detail::BoundFlagRefBase, 51  
Catch::Clara::Detail::BoundLambda< L >, 52  
Catch::Clara::Detail::BoundManyLambda< L >, 54  
Catch::Clara::Detail::BoundRef, 56  
Catch::Clara::Detail::BoundValueRef< std::vector< T >  
  >, 59  
Catch::Clara::Detail::BoundValueRef< T >, 57  
Catch::Clara::Detail::BoundValueRefBase, 60  
Catch::Clara::Detail::ComposableParserImpl< DerivedT  
  >, 74  
Catch::Clara::Detail::fake\_arg, 107  
Catch::Clara::Detail::HelpColumns, 121  
Catch::Clara::Detail::is\_unary\_function< F, Catch::Detail::void\_t<  
  decltype(std::declval< F >())(fake\_arg())> >  
  >, 140  
Catch::Clara::Detail::is\_unary\_function< F, typename  
  >, 139  
Catch::Clara::Detail::LambdaInvoker< ReturnType >,  
  151  
Catch::Clara::Detail::LambdaInvoker< void >, 152  
Catch::Clara::Detail::ParserBase, 194  
Catch::Clara::Detail::ParserRefImpl< DerivedT >, 195  
Catch::Clara::Detail::ParseState, 196  
Catch::Clara::Detail::ResultBase, 218  
Catch::Clara::Detail::ResultValueBase< T >, 219  
Catch::Clara::Detail::ResultValueBase< void >, 221  
Catch::Clara::Detail::Token, 273  
Catch::Clara::Detail::TokenStream, 274  
Catch::Clara::Detail::UnaryLambdaTraits< L >, 281  
Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::\*)(Args...) const >,  
  282  
Catch::Clara::Detail::UnaryLambdaTraits< ReturnT(ClassT::\*)(ArgT) const >,  
  282  
Catch::Clara::ExeName, 106  
Catch::Clara::Help, 120  
Catch::Clara::Opt, 190  
Catch::Clara::Parser, 193  
Catch::Colour, 67

- Catch::ColourImpl, 68
  - guardColour, 69
- Catch::ColourImpl::ColourGuard, 67
  - engage, 68
- Catch::CompactReporter, 71
  - testRunEnded, 72
  - testRunStarting, 72
- Catch::Config, 75
- Catch::ConfigData, 76
- Catch::ConsoleReporter, 79
  - testRunEnded, 80
  - testRunStarting, 81
- Catch::Counts, 86
- Catch::CumulativeReporterBase, 87
  - testRunEnded, 89
  - testRunStarting, 89
- Catch::CumulativeReporterBase::Node< T, ChildNodeT >, 186
- Catch::CumulativeReporterBase::SectionNode, 229
- Catch::Decomposer, 92
- Catch::Detail::AssertionOrBenchmarkResult, 35
- Catch::Detail::CaseInsensitiveEqualTo, 62
- Catch::Detail::CaseInsensitiveLess, 62
- Catch::Detail::EnumInfo, 94
- Catch::Detail::EnumValuesRegistry, 95
- Catch::Detail::has\_description< T, typename >, 117
- Catch::Detail::has\_description< T, void\_t< decltype(T::get\_description) > >, 118
- Catch::Detail::is\_range\_impl< T, typename >, 136
- Catch::Detail::is\_range\_impl< T, void\_t< decltype(begin(std::declval< T >())) > >, 137
- Catch::Detail::IsStreamInsertable< T >, 143
- Catch::Detail::make\_void<... >, 155
- Catch::Detail::NonCopyable, 187
- Catch::Detail::unique\_ptr< T >, 282
- Catch::ErrnoGuard, 99
- Catch::EventListenerBase, 100
  - testRunEnded, 102
  - testRunStarting, 102
- Catch::EventListenerFactory, 102
- Catch::ExceptionTranslatorRegistrar, 104
- Catch::ExceptionTranslatorRegistry, 104
- Catch::ExprLhs< LhsT >, 107
- Catch::FatalConditionHandler, 108
- Catch::FatalConditionHandlerGuard, 108
- Catch::GeneratorException, 112
- Catch::Generators::as< T >, 34
- Catch::Generators::ChunkGenerator< T >, 65
  - next, 66
- Catch::Generators::FilterGenerator< T, Predicate >, 109
  - next, 110
- Catch::Generators::FixedValuesGenerator< T >, 111
  - next, 112
- Catch::Generators::Generators< T >, 113
  - next, 114
- Catch::Generators::GeneratorUntypedBase, 115
  - countedNext, 115
  - currentElementAsString, 116
- Catch::Generators::GeneratorWrapper< T >, 116
- Catch::Generators::IGenerator< T >, 128
- Catch::Generators::IteratorGenerator< T >, 145
  - next, 146
- Catch::Generators::MapGenerator< T, U, Func >, 155
  - next, 156
- Catch::Generators::RandomFloatingGenerator< Float >, 200
  - next, 201
- Catch::Generators::RandomIntegerGenerator< Integer >, 202
  - next, 203
- Catch::Generators::RangeGenerator< T >, 203
  - next, 204
- Catch::Generators::RepeatGenerator< T >, 210
  - next, 211
- Catch::Generators::SingleValueGenerator< T >, 235
  - next, 236
- Catch::Generators::TakeGenerator< T >, 259
  - next, 260
- Catch::IConfig, 122
- Catch::IContext, 123
- Catch::IEventListener, 124
  - testRunEnded, 126
  - testRunStarting, 126
- Catch::IExceptionTranslator, 127
- Catch::IExceptionTranslatorRegistry, 127
- Catch::IGeneratorTracker, 129
- Catch::IMutableContext, 129
- Catch::IMutableEnumValuesRegistry, 130
- Catch::IMutableRegistryHub, 131
- Catch::IRegistryHub, 131
- Catch::IReporterFactory, 131
- Catch::IReporterRegistry, 132
- Catch::IResultCapture, 133
- Catch::is\_callable< Fun(Args...) >, 134
- Catch::is\_callable< T >, 134
- Catch::is\_callable\_tester, 135
- Catch::is\_range< T >, 135
- Catch::ISingleton, 142
- Catch::IStream, 143
  - isConsole, 143
- Catch::ITagAliasRegistry, 144
- Catch::ITestCaseRegistry, 146
- Catch::ITestInvoker, 147
- Catch::ITransientExpression, 149
- Catch::JUnitReporter, 150
  - testRunStarting, 151
- Catch::LazyExpression, 152
- Catch::LeakDetector, 152
- Catch::lineOfChars, 154
- Catch::ListenerDescription, 154
- Catch::ListenerRegistrar< T >, 155
- Catch::Matchers::AllMatchMatcher< Matcher >, 27
- Catch::Matchers::AnyMatchMatcher< Matcher >, 29
- Catch::Matchers::ApproxMatcher< T, AllocComp, AllocMatch >, 31



- Catch::Matchers::CasedString, 62
- Catch::Matchers::ContainsElementMatcher< T, Equality >, 82
- Catch::Matchers::ContainsMatcher< T, AllocComp, AllocMatch >, 84
- Catch::Matchers::ContainsMatcherMatcher< Matcher >, 85
- Catch::Matchers::Detail::conjunction< Cond >, 77
- Catch::Matchers::Detail::conjunction< Cond, Rest... >, 78
- Catch::Matchers::Detail::MatchAllOf< ArgT >, 157
- Catch::Matchers::Detail::MatchAllOfGeneric< MatcherTs >, 158
- Catch::Matchers::Detail::MatchAnyOf< ArgT >, 160
- Catch::Matchers::Detail::MatchAnyOfGeneric< MatcherTs >, 161
- Catch::Matchers::Detail::MatchNotOf< ArgT >, 168
- Catch::Matchers::Detail::MatchNotOfGeneric< MatcherT >, 170
- Catch::Matchers::EndsWithMatcher, 93
- Catch::Matchers::EqualsMatcher< T, AllocComp, AllocMatch >, 98
- Catch::Matchers::ExceptionMessageMatcher, 103
- Catch::Matchers::HasSizeMatcher, 119
- Catch::Matchers::IsEmptyMatcher, 141
- Catch::Matchers::MatcherBase< T >, 163
- Catch::Matchers::MatcherGenericBase, 164
- Catch::Matchers::MatcherUntypedBase, 166
- Catch::Matchers::NoneMatchMatcher< Matcher >, 188
- Catch::Matchers::PredicateMatcher< T, Predicate >, 197
- Catch::Matchers::RegexMatcher, 208
- Catch::Matchers::SizeMatchesMatcher< Matcher >, 237
- Catch::Matchers::StartsWithMatcher, 240
- Catch::Matchers::StringContainsMatcher, 244
- Catch::Matchers::StringEqualsMatcher, 245
- Catch::Matchers::StringMatcherBase, 254
- Catch::Matchers::UnorderedEqualsMatcher< T, AllocComp, AllocMatch >, 283
- Catch::Matchers::VectorContainsElementMatcher< T, Alloc >, 287
- Catch::Matchers::WithinAbsMatcher, 293
- Catch::Matchers::WithinRelMatcher, 295
- Catch::Matchers::WithinUlpMatcher, 296
- Catch::MatchExpr< ArgT, MatcherT >, 167
- Catch::MessageBuilder, 171
- Catch::MessageInfo, 172
- Catch::MessageStream, 173
- Catch::MultiReporter, 182
  - testRunEnded, 184
  - testRunStarting, 184
- Catch::NameAndTags, 186
- Catch::Optional< T >, 192
- Catch::pluralise, 197
- Catch::ProcessedReporterSpec, 198
- Catch::ratio\_string< Ratio >, 205
- Catch::ratio\_string< std::atto >, 205
- Catch::ratio\_string< std::femto >, 205
- Catch::ratio\_string< std::micro >, 205
- Catch::ratio\_string< std::milli >, 206
- Catch::ratio\_string< std::nano >, 206
- Catch::ratio\_string< std::pico >, 206
- Catch::RedirectedStdErr, 206
- Catch::RedirectedStdOut, 207
- Catch::RedirectedStream, 207
- Catch::RedirectedStreams, 207
- Catch::RegistrarForTagAliases, 209
- Catch::ReporterBase, 211
  - listListeners, 212
  - listReporters, 212
  - listTags, 213
  - listTests, 213
  - m\_stream, 213
- Catch::ReporterConfig, 214
- Catch::ReporterDescription, 214
- Catch::ReporterFactory< T >, 215
- Catch::ReporterPreferences, 215
  - shouldRedirectStdOut, 216
  - shouldReportAllAssertions, 216
- Catch::ReporterRegistrar< T >, 216
- Catch::ReporterRegistry, 217
- Catch::ReporterSpec, 218
- Catch::ResultDisposition, 219
- Catch::ResultWas, 222
- Catch::ReusableStringStream, 222
- Catch::RunContext, 223
- Catch::ScopedMessage, 226
- Catch::Section, 227
- Catch::SectionEndInfo, 228
- Catch::SectionInfo, 228
- Catch::SectionStats, 230
- Catch::Session, 232
- Catch::SimplePcg32, 233
- Catch::Singleton< SingletonImplT, InterfaceT, MutableInterfaceT >, 234
- Catch::SonarQubeReporter, 238
  - testRunStarting, 239
- Catch::SourceLineInfo, 239
- Catch::StartupExceptionRegistry, 241
- Catch::StreamEndStop, 241
- Catch::StreamingReporterBase, 241
  - testRunEnded, 242
  - testRunStarting, 243
- Catch::StringMaker< bool >, 246
- Catch::StringMaker< Catch::Approx >, 246
- Catch::StringMaker< char >, 247
- Catch::StringMaker< char \* >, 246
- Catch::StringMaker< char const \* >, 247
- Catch::StringMaker< char[SZ]>, 247
- Catch::StringMaker< double >, 247
- Catch::StringMaker< float >, 248
- Catch::StringMaker< int >, 248
- Catch::StringMaker< long >, 248
- Catch::StringMaker< long long >, 248
- Catch::StringMaker< R C::\* >, 249

- Catch::StringMaker< R, std::enable\_if\_t< is\_range< R  
>::value &&!::Catch::Detail::IsStreamInsertable< R  
>::value >, 249
- Catch::StringMaker< signed char >, 249
- Catch::StringMaker< signed char[SZ]>, 249
- Catch::StringMaker< std::chrono::duration< Value, Ra-  
tio > >, 250
- Catch::StringMaker< std::chrono::duration< Value,  
std::ratio< 1 > >, 250
- Catch::StringMaker< std::chrono::duration< Value,  
std::ratio< 3600 > >, 250
- Catch::StringMaker< std::chrono::duration< Value,  
std::ratio< 60 > >, 250
- Catch::StringMaker< std::chrono::time\_point< Clock,  
Duration > >, 251
- Catch::StringMaker< std::chrono::time\_point< std::chrono:  
system\_clock, Duration > >, 251
- Catch::StringMaker< std::nullptr\_t >, 251
- Catch::StringMaker< std::string >, 251
- Catch::StringMaker< std::wstring >, 252
- Catch::StringMaker< T \* >, 252
- Catch::StringMaker< T, typename >, 246
- Catch::StringMaker< T[SZ]>, 252
- Catch::StringMaker< unsigned char >, 252
- Catch::StringMaker< unsigned char[SZ]>, 253
- Catch::StringMaker< unsigned int >, 253
- Catch::StringMaker< unsigned long >, 253
- Catch::StringMaker< unsigned long long >, 253
- Catch::StringMaker< wchar\_t \* >, 254
- Catch::StringMaker< wchar\_t const \* >, 254
- Catch::StringRef, 255
  - compare, 256
- Catch::Tag, 256
- Catch::TagAlias, 257
- Catch::TagAliasRegistry, 258
- Catch::TagInfo, 258
- Catch::TAPReporter, 261
  - testRunEnded, 262
  - testRunStarting, 262
- Catch::TeamCityReporter, 263
  - testRunEnded, 264
  - testRunStarting, 264
- Catch::TestCaseHandle, 264
- Catch::TestCaseInfo, 265
- Catch::TestCaseInfoHasher, 266
- Catch::TestCaseStats, 267
- Catch::TestCaseTracking::ITracker, 147
  - findChild, 149
  - isGeneratorTracker, 149
  - isSectionTracker, 149
- Catch::TestCaseTracking::NameAndLocation, 185
- Catch::TestCaseTracking::SectionTracker, 231
  - isSectionTracker, 232
- Catch::TestCaseTracking::TrackerBase, 278
- Catch::TestCaseTracking::TrackerContext, 279
- Catch::TestFailureException, 267
- Catch::TestInvokerAsFunction, 268
- Catch::TestInvokerAsMethod< C >, 269
- Catch::TestRegistry, 270
- Catch::TestRunInfo, 271
- Catch::TestRunStats, 271
- Catch::TestSpec, 272
- Catch::TestSpec::FilterMatch, 110
- Catch::TestSpecParser, 273
- Catch::TextFlow::Column, 69
- Catch::TextFlow::Column::const\_iterator, 81
- Catch::TextFlow::Columns, 70
- Catch::TextFlow::Columns::iterator, 144
- Catch::Timer, 273
- Catch::Totals, 277
- Catch::true\_given< typename >, 280
- Catch::UnaryExpr< LhsT >, 281
- Catch::Version, 288
- Catch::WaitForKeypress, 292
- Catch::WarnAbout, 292
  - NoAssertions, 293
  - UnmatchedTestSpec, 293
  - What, 293
- Catch::WildcardPattern, 293
- Catch::XmlEncode, 297
- Catch::XmlReporter, 298
  - listListeners, 299
  - listReporters, 299
  - listTags, 299
  - listTests, 300
  - testRunEnded, 300
  - testRunStarting, 300
- Catch::XmlWriter, 301
  - writeAttribute, 301
- Catch::XmlWriter::ScopedElement, 225
- catch\_amalgamated.hpp
  - ANSI, 345
  - CATCH\_INTERNAL\_DEFINE\_EXPRESSION\_OPERATOR,  
331
  - CATCH\_REGISTER\_LISTENER, 331
  - CATCH\_REGISTER\_REPORTER, 331
  - CATCH\_REGISTER\_TAG\_ALIAS, 331
  - ColourMode, 345
  - Contains, 346
  - convertIntoString, 346
  - Default, 345
  - defaultListListeners, 346
  - defaultListReporters, 346
  - defaultListTags, 347
  - defaultListTests, 347
  - GENERATE, 332
  - GENERATE\_COPY, 332
  - GENERATE\_REF, 332
  - GenerateFrom, 345
  - INTERNAL\_CATCH\_BENCHMARK, 332
  - INTERNAL\_CATCH\_BENCHMARK\_ADVANCED,  
333
  - INTERNAL\_CATCH\_CAPTURE, 333
  - INTERNAL\_CATCH\_DECLARE\_SIG\_TEST1, 333
  - INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_METHOD1,  
333

- INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_METHOD\_X, [RuntimeError](#), [346](#)
- [334](#)
- INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_X, [ulpDistance](#), [349](#)
- [334](#) [Win32](#), [345](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST1, [Catch\\_global\\_namespace\\_dummy](#), [63](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_METHOD1, [CATCH\\_INTERNAL\\_DEFINE\\_EXPRESSION\\_OPERATOR](#)
- [334](#) [catch\\_amalgamated.hpp](#), [331](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_METHOD\_X, [CATCH\\_REGISTER\\_LISTENER](#)
- [335](#) [catch\\_amalgamated.hpp](#), [331](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_X, [CATCH\\_REGISTER\\_REPORTER](#)
- [335](#) [catch\\_amalgamated.hpp](#), [331](#)
- INTERNAL\_CATCH\_DYNAMIC\_SECTION, [CATCH\\_REGISTER\\_TAG\\_ALIAS](#)
- [335](#) [catch\\_amalgamated.hpp](#), [331](#)
- INTERNAL\_CATCH\_ELSE, [ColourMode](#)
- [335](#) [catch\\_amalgamated.hpp](#), [345](#)
- INTERNAL\_CATCH\_IF, [compare](#)
- [336](#) [Catch::StringRef](#), [256](#)
- INTERNAL\_CATCH\_METHOD\_AS\_TEST\_CASE, [Contains](#)
- [336](#) [catch\\_amalgamated.hpp](#), [346](#)
- INTERNAL\_CATCH\_MSG, [convertIntoString](#)
- [336](#) [catch\\_amalgamated.hpp](#), [346](#)
- INTERNAL\_CATCH\_NO\_THROW, [countedNext](#)
- [337](#) [Catch::Generators::GeneratorUntypedBase](#), [115](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER, [currentElementAsString](#)
- [337](#) [Catch::Generators::GeneratorUntypedBase](#), [116](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER0, [DataBase< TSeq >](#), [90](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER\_METHOD, [record\\_variant](#), [91](#)
- [338](#) [reproductive\\_number](#), [91](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER\_METHOD0, [Default](#)
- [338](#) [catch\\_amalgamated.hpp](#), [345](#)
- INTERNAL\_CATCH\_REGISTER\_ENUM, [defaultListListeners](#)
- [338](#) [catch\\_amalgamated.hpp](#), [346](#)
- INTERNAL\_CATCH\_REGISTER\_TESTCASE, [defaultListReporters](#)
- [339](#) [catch\\_amalgamated.hpp](#), [346](#)
- INTERNAL\_CATCH\_SECTION, [defaultListTags](#)
- [339](#) [catch\\_amalgamated.hpp](#), [347](#)
- INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_2, [defaultListTests](#)
- [339](#) [catch\\_amalgamated.hpp](#), [347](#)
- INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_METHOD\_2, [engage](#)
- [340](#) [Catch::ColourImpl::ColourGuard](#), [68](#)
- INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_2, [Entity< TSeq >](#), [94](#)
- [340](#)
- INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_2, [findChild](#)
- [341](#) [Catch::TestCaseTracking::ITracker](#), [149](#)
- INTERNAL\_CATCH\_TEST, [GENERATE](#)
- [341](#) [catch\\_amalgamated.hpp](#), [332](#)
- INTERNAL\_CATCH\_TEST\_CASE\_METHOD2, [GENERATE\\_COPY](#)
- [342](#) [catch\\_amalgamated.hpp](#), [332](#)
- INTERNAL\_CATCH\_TESTCASE2, [GENERATE\\_REF](#)
- [342](#) [catch\\_amalgamated.hpp](#), [332](#)
- INTERNAL\_CATCH\_THROWS, [GenerateFrom](#)
- [342](#) [catch\\_amalgamated.hpp](#), [345](#)
- INTERNAL\_CATCH\_THROWS\_AS, [guardColour](#)
- [343](#) [Catch::ColourImpl](#), [69](#)
- INTERNAL\_CATCH\_THROWS\_MATCHES, [include/catch2/catch\\_amalgamated.hpp](#), [303](#)
- [343](#) [INTERNAL\\_CATCH\\_BENCHMARK](#)
- INTERNAL\_CATCH\_THROWS\_STR\_MATCHES, [344](#)
- [344](#)
- INTERNAL\_CATCH\_TRANSLATE\_EXCEPTION2, [344](#)
- [344](#)
- INTERNAL\_CHECK\_THAT, [344](#)
- [344](#)
- LogicError, [346](#)
- makeStream, [347](#)
- None, [345](#)
- Ok, [346](#)
- operator&&, [348](#)
- operator| |, [348](#)
- parseReporterSpec, [348](#)
- PlatformDefault, [345](#)
- Predicate, [349](#)
- registerReporterImpl, [349](#)
- ResultType, [345](#)

- catch\_amalgamated.hpp, [332](#)
- INTERNAL\_CATCH\_BENCHMARK\_ADVANCED
  - catch\_amalgamated.hpp, [333](#)
- INTERNAL\_CATCH\_CAPTURE
  - catch\_amalgamated.hpp, [333](#)
- INTERNAL\_CATCH\_DECLARE\_SIG\_TEST1
  - catch\_amalgamated.hpp, [333](#)
- INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_METHOD1
  - catch\_amalgamated.hpp, [333](#)
- INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_METHOD\_X
  - catch\_amalgamated.hpp, [334](#)
- INTERNAL\_CATCH\_DECLARE\_SIG\_TEST\_X
  - catch\_amalgamated.hpp, [334](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST1
  - catch\_amalgamated.hpp, [334](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_METHOD1
  - catch\_amalgamated.hpp, [334](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_METHOD\_X
  - catch\_amalgamated.hpp, [335](#)
- INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_X
  - catch\_amalgamated.hpp, [335](#)
- INTERNAL\_CATCH\_DYNAMIC\_SECTION
  - catch\_amalgamated.hpp, [335](#)
- INTERNAL\_CATCH\_ELSE
  - catch\_amalgamated.hpp, [335](#)
- INTERNAL\_CATCH\_IF
  - catch\_amalgamated.hpp, [336](#)
- INTERNAL\_CATCH\_METHOD\_AS\_TEST\_CASE
  - catch\_amalgamated.hpp, [336](#)
- INTERNAL\_CATCH\_MSG
  - catch\_amalgamated.hpp, [336](#)
- INTERNAL\_CATCH\_NO\_THROW
  - catch\_amalgamated.hpp, [336](#)
- INTERNAL\_CATCH\_NTTP\_1
  - catch\_amalgamated.hpp, [337](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER
  - catch\_amalgamated.hpp, [337](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER0
  - catch\_amalgamated.hpp, [337](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER\_METHOD
  - catch\_amalgamated.hpp, [338](#)
- INTERNAL\_CATCH\_NTTP\_REGISTER\_METHOD0
  - catch\_amalgamated.hpp, [338](#)
- INTERNAL\_CATCH\_REGISTER\_ENUM
  - catch\_amalgamated.hpp, [338](#)
- INTERNAL\_CATCH\_REGISTER\_TESTCASE
  - catch\_amalgamated.hpp, [338](#)
- INTERNAL\_CATCH\_SECTION
  - catch\_amalgamated.hpp, [339](#)
- INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_2
  - catch\_amalgamated.hpp, [339](#)
- INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_METHOD\_2
  - catch\_amalgamated.hpp, [339](#)
- INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_2
  - catch\_amalgamated.hpp, [340](#)
- INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_2
  - catch\_amalgamated.hpp, [341](#)
- INTERNAL\_CATCH\_TEST
  - catch\_amalgamated.hpp, [341](#)
- INTERNAL\_CATCH\_TEST\_CASE\_METHOD2
  - catch\_amalgamated.hpp, [342](#)
- INTERNAL\_CATCH\_TESTCASE2
  - catch\_amalgamated.hpp, [342](#)
- INTERNAL\_CATCH\_THROWS
  - catch\_amalgamated.hpp, [342](#)
- INTERNAL\_CATCH\_THROWS\_AS
  - catch\_amalgamated.hpp, [343](#)
- INTERNAL\_CATCH\_THROWS\_MATCHES
  - catch\_amalgamated.hpp, [343](#)
- INTERNAL\_CATCH\_THROWS\_STR\_MATCHES
  - catch\_amalgamated.hpp, [344](#)
- INTERNAL\_CATCH\_TRANSLATE\_EXCEPTION2
  - catch\_amalgamated.hpp, [344](#)
- INTERNAL\_CHECK\_THAT
  - catch\_amalgamated.hpp, [344](#)
- isConsole
  - Catch::IStream, [143](#)
- isGeneratorTracker
  - Catch::TestCaseTracking::ITracker, [149](#)
- isSectionTracker
  - Catch::TestCaseTracking::ITracker, [149](#)
  - Catch::TestCaseTracking::SectionTracker, [232](#)
- LFMCMC< TData >, [152](#)
- listListeners
  - Catch::ReporterBase, [212](#)
  - Catch::XmlReporter, [299](#)
- listReporters
  - Catch::ReporterBase, [212](#)
  - Catch::XmlReporter, [299](#)
- listTags
  - Catch::ReporterBase, [213](#)
  - Catch::XmlReporter, [299](#)
- listTests
  - Catch::ReporterBase, [213](#)
  - Catch::XmlReporter, [300](#)
- LogicError
  - catch\_amalgamated.hpp, [346](#)
- m\_stream
  - Catch::ReporterBase, [213](#)
- makeStream
  - catch\_amalgamated.hpp, [347](#)
- Model< TSeq >, [174](#)
  - add\_global\_action, [180](#)
  - reset, [181](#)
  - run\_multiple, [181](#)
  - write\_data, [181](#)
- next
  - Catch::Generators::ChunkGenerator< T >, [66](#)
  - Catch::Generators::FilterGenerator< T, Predicate >, [110](#)
  - Catch::Generators::FixedValuesGenerator< T >, [112](#)
  - Catch::Generators::Generators< T >, [114](#)
  - Catch::Generators::IteratorGenerator< T >, [146](#)

- Catch::Generators::MapGenerator< T, U, Func >, [156](#)
- Catch::Generators::RandomFloatingGenerator< Float >, [201](#)
- Catch::Generators::RandomIntegerGenerator< Integer >, [203](#)
- Catch::Generators::RangeGenerator< T >, [204](#)
- Catch::Generators::RepeatGenerator< T >, [211](#)
- Catch::Generators::SingleValueGenerator< T >, [236](#)
- Catch::Generators::TakeGenerator< T >, [260](#)
- NoAssertions
  - Catch::WarnAbout, [293](#)
- None
  - catch\_amalgamated.hpp, [345](#)
- Ok
  - catch\_amalgamated.hpp, [346](#)
- operator&&
  - catch\_amalgamated.hpp, [348](#)
- operator| |
  - catch\_amalgamated.hpp, [348](#)
- parseReporterSpec
  - catch\_amalgamated.hpp, [348](#)
- PersonTools< TSeq >, [196](#)
- PlatformDefault
  - catch\_amalgamated.hpp, [345](#)
- Predicate
  - catch\_amalgamated.hpp, [349](#)
- Progress, [199](#)
- Queue< TSeq >, [199](#)
- RandGraph, [200](#)
- read\_edgelist
  - AdjList, [24](#)
- record\_variant
  - DataBase< TSeq >, [91](#)
- registerReporterImpl
  - catch\_amalgamated.hpp, [349](#)
- reproductive\_number
  - DataBase< TSeq >, [91](#)
- reset
  - Model< TSeq >, [181](#)
- ResultType
  - catch\_amalgamated.hpp, [345](#)
- run\_multiple
  - Model< TSeq >, [181](#)
- RuntimeError
  - catch\_amalgamated.hpp, [346](#)
- shouldRedirectStdOut
  - Catch::ReporterPreferences, [216](#)
- shouldReportAllAssertions
  - Catch::ReporterPreferences, [216](#)
- testRunEnded
  - Catch::CompactReporter, [72](#)
  - Catch::ConsoleReporter, [80](#)
  - Catch::CumulativeReporterBase, [89](#)
  - Catch::EventListenerBase, [102](#)
  - Catch::EventListener, [126](#)
  - Catch::MultiReporter, [184](#)
  - Catch::StreamingReporterBase, [242](#)
  - Catch::TAPReporter, [262](#)
  - Catch::TeamCityReporter, [264](#)
  - Catch::XmlReporter, [300](#)
- testRunStarting
  - Catch::CompactReporter, [72](#)
  - Catch::ConsoleReporter, [81](#)
  - Catch::CumulativeReporterBase, [89](#)
  - Catch::EventListenerBase, [102](#)
  - Catch::EventListener, [126](#)
  - Catch::JUnitReporter, [151](#)
  - Catch::MultiReporter, [184](#)
  - Catch::SonarQubeReporter, [239](#)
  - Catch::StreamingReporterBase, [243](#)
  - Catch::TAPReporter, [262](#)
  - Catch::TeamCityReporter, [264](#)
  - Catch::XmlReporter, [300](#)
- Tool< TSeq >, [274](#)
- Tools< TSeq >, [275](#)
- Tools\_const< TSeq >, [276](#)
- ulpDistance
  - catch\_amalgamated.hpp, [349](#)
- UnmatchedTestSpec
  - Catch::WarnAbout, [293](#)
- UserData
  - UserData< TSeq >, [286](#)
- UserData< TSeq >, [284](#)
- UserData, [286](#)
- vecHasher< T >, [286](#)
- Virus< TSeq >, [289](#)
- Viruses< TSeq >, [291](#)
- Viruses\_const< TSeq >, [291](#)
- What
  - Catch::WarnAbout, [293](#)
- Win32
  - catch\_amalgamated.hpp, [345](#)
- write\_data
  - Model< TSeq >, [181](#)
- writeAttribute
  - Catch::XmlWriter, [301](#)