

yaramod-v4

Generated by Doxygen 1.9.1

1 Documentation main page	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 File Index	13
4.1 File List	13
5 Class Documentation	17
5.1 AddExpression Class Reference	17
5.1.1 Member Function Documentation	18
5.1.1.1 accept()	18
5.1.1.2 opsign()	19
5.2 AllExpression Class Reference	19
5.2.1 Detailed Description	20
5.2.2 Member Function Documentation	20
5.2.2.1 accept()	20
5.2.2.2 getTextFormatted()	21
5.3 AndExpression Class Reference	21
5.3.1 Member Function Documentation	22
5.3.1.1 accept()	22
5.3.1.2 opsign()	23
5.4 AnyExpression Class Reference	23
5.4.1 Detailed Description	24
5.4.2 Member Function Documentation	24
5.4.2.1 accept()	24
5.4.2.2 getTextFormatted()	25
5.5 ArrayExpression Class Reference	25
5.5.1 Member Function Documentation	26
5.5.1.1 accept()	26
5.5.1.2 areOperandsValid()	27
5.5.1.3 getTextFormatted()	27
5.5.1.4 isComplete()	27
5.5.1.5 isValid()	27
5.6 AtExpression Class Reference	28
5.6.1 Member Function Documentation	29
5.6.1.1 accept()	29
5.6.1.2 getType()	30
5.6.1.3 isValid()	30
5.6.1.4 opsign()	30
5.7 BinaryArithmeticExpression Class Reference	31

5.7.1 Detailed Description	32
5.7.2 Member Function Documentation	32
5.7.2.1 getType()	32
5.7.2.2 isValid()	32
5.8 BinaryBitwiseExpression Class Reference	33
5.8.1 Detailed Description	34
5.8.2 Member Function Documentation	34
5.8.2.1 getType()	34
5.8.2.2 isValid()	34
5.9 BinaryBoolExpression Class Reference	35
5.9.1 Detailed Description	36
5.9.2 Member Function Documentation	36
5.9.2.1 getType()	36
5.9.2.2 isValid()	37
5.10 BinaryExpression Class Reference	37
5.10.1 Detailed Description	39
5.10.2 Member Function Documentation	39
5.10.2.1 areOperandsValid()	40
5.10.2.2 getTextFormatted()	40
5.10.2.3 isComplete()	40
5.10.2.4 opsign()	40
5.11 BinaryRelationalExpression Class Reference	41
5.11.1 Detailed Description	42
5.11.2 Member Function Documentation	42
5.11.2.1 getType()	42
5.11.2.2 isValid()	42
5.12 BinaryStrRelationalExpression Class Reference	43
5.12.1 Detailed Description	44
5.12.2 Member Function Documentation	44
5.12.2.1 getType()	44
5.12.2.2 isValid()	44
5.13 BitwiseAndExpression Class Reference	45
5.13.1 Member Function Documentation	46
5.13.1.1 accept()	46
5.13.1.2 opsign()	47
5.14 BitwiseNotExpression Class Reference	47
5.14.1 Member Function Documentation	48
5.14.1.1 accept()	49
5.15 BitwiseOrExpression Class Reference	49
5.15.1 Member Function Documentation	50
5.15.1.1 accept()	50
5.15.1.2 opsign()	51

5.16 BitwiseXorExpression Class Reference	51
5.16.1 Member Function Documentation	52
5.16.1.1 accept()	52
5.16.1.2 opsign()	53
5.17 FileContext::cache_entry_t Struct Reference	53
5.17.1 Detailed Description	53
5.18 FileContext::cache_t Struct Reference	54
5.18.1 Detailed Description	54
5.19 Comment Class Reference	55
5.19.1 Detailed Description	56
5.19.2 Member Function Documentation	56
5.19.2.1 getTextFormatted()	56
5.20 ContainsExpression Class Reference	57
5.20.1 Member Function Documentation	58
5.20.1.1 accept()	58
5.20.1.2 opsign()	59
5.21 DefinedExpression Class Reference	59
5.21.1 Member Function Documentation	60
5.21.1.1 accept()	60
5.21.1.2 getType()	61
5.21.1.3 isValid()	61
5.22 DivExpression Class Reference	62
5.22.1 Member Function Documentation	63
5.22.1.1 accept()	63
5.22.1.2 opsign()	64
5.23 edit_t Struct Reference	64
5.23.1 Detailed Description	64
5.24 EndsWithExpression Class Reference	65
5.24.1 Member Function Documentation	66
5.24.1.1 accept()	66
5.24.1.2 opsign()	67
5.25 Enum Class Reference	67
5.25.1 Member Function Documentation	68
5.25.1.1 accept()	68
5.25.1.2 areOperandsValid()	69
5.25.1.3 getTextFormatted()	69
5.25.1.4 getType()	69
5.25.1.5 isComplete()	69
5.25.1.6 isValid()	69
5.26 EqExpression Class Reference	70
5.26.1 Member Function Documentation	71
5.26.1.1 accept()	71

5.26.1.2 opsign()	72
5.27 Error Class Reference	72
5.27.1 Detailed Description	73
5.27.2 Member Function Documentation	73
5.27.2.1 exception()	73
5.27.2.2 getDescription()	73
5.28 ErrorCollector< E > Class Template Reference	73
5.28.1 Detailed Description	74
5.28.2 Member Function Documentation	74
5.28.2.1 add()	74
5.28.2.2 clear()	74
5.28.2.3 dump()	75
5.28.2.4 get()	75
5.28.2.5 getConst()	75
5.28.2.6 remove()	75
5.28.2.7 revRemoveUntil()	75
5.28.2.8 update()	76
5.29 ErrorException Class Reference	76
5.29.1 Detailed Description	77
5.30 Expression Class Reference	77
5.30.1 Detailed Description	80
5.30.2 Member Function Documentation	80
5.30.2.1 accept()	80
5.30.2.2 areOperandsValid()	80
5.30.2.3 getType()	81
5.30.2.4 isArithmeticCompatible()	81
5.30.2.5 isComplete()	81
5.30.2.6 isValid()	81
5.30.2.7 typeToString()	82
5.31 ExpressionError Class Reference	82
5.31.1 Detailed Description	83
5.32 ExpressionErrorException Class Reference	83
5.32.1 Detailed Description	84
5.33 ExpressionSetExpression Class Reference	85
5.33.1 Detailed Description	86
5.33.2 Member Function Documentation	86
5.33.2.1 accept()	87
5.33.2.2 getTextFormatted()	87
5.34 ExtVariable Class Reference	87
5.34.1 Detailed Description	88
5.34.2 Member Function Documentation	88
5.34.2.1 getValue()	88

5.34.2.2 setValue()	88
5.35 FileContext Class Reference	89
5.35.1 Detailed Description	89
5.35.2 Member Function Documentation	89
5.35.2.1 clearTree()	89
5.35.2.2 diffTree()	90
5.35.2.3 getString()	90
5.35.2.4 getTempString()	90
5.35.2.5 getTree()	90
5.35.2.6 setString()	90
5.35.2.7 setTree()	90
5.35.2.8 stringEdit()	91
5.35.2.9 stringUpdate()	91
5.35.2.10 toTSEdit() [1/2]	91
5.35.2.11 toTSEdit() [2/2]	91
5.35.2.12 treeEdit()	91
5.35.2.13 undoEdits()	91
5.36 FileInclude Class Reference	92
5.36.1 Detailed Description	93
5.36.2 Member Function Documentation	93
5.36.2.1 getFile()	93
5.36.2.2 getId()	93
5.36.2.3 getPath()	93
5.36.2.4 getTextFormatted()	93
5.37 ForExpression Class Reference	94
5.37.1 Detailed Description	95
5.37.2 Member Function Documentation	95
5.37.2.1 accept()	95
5.37.2.2 areOperandsValid()	95
5.37.2.3 getTextFormatted()	96
5.37.2.4 getType()	96
5.37.2.5 isComplete()	96
5.37.2.6 isValid()	96
5.38 ForIntExpression Class Reference	97
5.38.1 Detailed Description	98
5.38.2 Member Function Documentation	99
5.38.2.1 accept()	99
5.38.2.2 areOperandsValid()	99
5.38.2.3 getTextFormatted()	99
5.38.2.4 getType()	99
5.38.2.5 isComplete()	100
5.38.2.6 isValid()	100

5.39 FunctionCallError Class Reference	100
5.39.1 Detailed Description	102
5.40 FunctionCallErrorException Class Reference	103
5.40.1 Detailed Description	104
5.41 FunctionCallExpression Class Reference	105
5.41.1 Member Function Documentation	106
5.41.1.1 accept()	107
5.41.1.2 areOperandsValid()	107
5.41.1.3 getArgs()	107
5.41.1.4 getTextFormatted()	107
5.41.1.5 getType()	108
5.41.1.6 hasValidArgs()	108
5.41.1.7 isComplete()	108
5.41.1.8 isValid()	108
5.42 GteExpression Class Reference	109
5.42.1 Member Function Documentation	110
5.42.1.1 accept()	110
5.42.1.2 opsign()	111
5.43 GtExpression Class Reference	111
5.43.1 Member Function Documentation	112
5.43.1.1 accept()	112
5.43.1.2 opsign()	113
5.44 HexStrError Class Reference	113
5.44.1 Detailed Description	114
5.45 HexStrErrorException Class Reference	115
5.45.1 Detailed Description	116
5.46 IContainsExpression Class Reference	117
5.46.1 Member Function Documentation	118
5.46.1.1 accept()	118
5.46.1.2 opsign()	119
5.47 IEndsWithExpression Class Reference	119
5.47.1 Member Function Documentation	120
5.47.1.1 accept()	120
5.47.1.2 opsign()	121
5.48 IEqExpression Class Reference	121
5.48.1 Member Function Documentation	122
5.48.1.1 accept()	122
5.48.1.2 opsign()	123
5.49 Import Class Reference	123
5.49.1 Detailed Description	124
5.49.2 Member Function Documentation	124
5.49.2.1 getIdx()	124

5.49.2.2 getTextFormatted()	125
5.50 IncludeError Class Reference	125
5.50.1 Detailed Description	126
5.51 IncludeErrorException Class Reference	126
5.51.1 Detailed Description	128
5.52 IndexError Class Reference	128
5.52.1 Detailed Description	130
5.53 IndexErrorException Class Reference	131
5.53.1 Detailed Description	132
5.54 InExpression Class Reference	133
5.54.1 Detailed Description	134
5.54.2 Member Function Documentation	134
5.54.2.1 accept()	134
5.54.2.2 areOperandsValid()	134
5.54.2.3 getTextFormatted()	135
5.54.2.4 getType()	135
5.54.2.5 isComplete()	135
5.54.2.6 isValid()	135
5.55 InternalErrorException Class Reference	136
5.55.1 Detailed Description	137
5.56 IntVariable Class Reference	137
5.56.1 Detailed Description	138
5.56.2 Member Function Documentation	138
5.56.2.1 getTextFormatted()	138
5.56.2.2 getValue()	139
5.56.2.3 setValue()	139
5.57 IParserAdapter Class Reference	139
5.57.1 Detailed Description	140
5.57.2 Member Function Documentation	140
5.57.2.1 createYaraSource()	140
5.57.2.2 createYaraSourceFromFile()	140
5.57.2.3 updateYaraSource()	141
5.57.2.4 updateYaraSourceFromFile()	141
5.58 IStartsWithExpression Class Reference	142
5.58.1 Member Function Documentation	143
5.58.1.1 accept()	143
5.58.1.2 opsign()	144
5.59 RuleModifierContainer::iterator Class Reference	144
5.59.1 Detailed Description	144
5.60 StringModifierContainer::iterator Class Reference	145
5.60.1 Detailed Description	145
5.61 SymTab< V >::iterator Class Reference	145

5.61.1 Detailed Description	146
5.62 SymTabDuplId< V >::iterator Class Reference	146
5.62.1 Detailed Description	147
5.63 SymTabDuplId< V >::Pool::iterator Class Reference	147
5.63.1 Detailed Description	148
5.64 SymTabOffset< V >::iterator Class Reference	148
5.64.1 Detailed Description	149
5.65 LeftShiftExpression Class Reference	149
5.65.1 Member Function Documentation	150
5.65.1.1 accept()	150
5.65.1.2 opsign()	151
5.66 Literal Class Reference	151
5.66.1 Detailed Description	152
5.66.2 Member Function Documentation	152
5.66.2.1 get()	152
5.66.2.2 getRaw()	152
5.66.2.3 getTextFormatted()	152
5.66.2.4 isType()	153
5.66.2.5 isUndefined()	153
5.66.2.6 set()	153
5.67 LiteralExpression Class Reference	153
5.67.1 Detailed Description	155
5.67.2 Member Function Documentation	155
5.67.2.1 accept()	156
5.67.2.2 getTextFormatted()	156
5.67.2.3 getType()	156
5.67.2.4 isComplete()	156
5.68 LteExpression Class Reference	157
5.68.1 Member Function Documentation	158
5.68.1.1 accept()	158
5.68.1.2 opsign()	159
5.69 LtExpression Class Reference	159
5.69.1 Member Function Documentation	160
5.69.1.1 accept()	160
5.69.1.2 opsign()	161
5.70 MatchesExpression Class Reference	161
5.70.1 Member Function Documentation	162
5.70.1.1 accept()	162
5.70.1.2 getType()	163
5.70.1.3 isValid()	163
5.70.1.4 opsign()	163
5.71 Meta Class Reference	164

5.71.1 Detailed Description	165
5.71.2 Member Function Documentation	165
5.71.2.1 getId()	165
5.71.2.2 getTextFormatted()	165
5.71.2.3 getValue()	165
5.71.2.4 setId()	165
5.71.2.5 setValue()	166
5.72 MissingTokenError Class Reference	166
5.72.1 Detailed Description	167
5.73 MissingTokenErrorException Class Reference	168
5.73.1 Detailed Description	169
5.74 Module Class Reference	169
5.74.1 Detailed Description	170
5.74.2 Member Function Documentation	170
5.74.2.1 buildModuleContext()	170
5.74.2.2 findField()	170
5.74.2.3 findInArray()	171
5.74.2.4 getData()	171
5.74.2.5 getDataType()	171
5.74.2.6 getElementDataType()	171
5.74.2.7 getElementType()	172
5.74.2.8 jsonToExprType()	172
5.74.2.9 load()	172
5.74.2.10 merge()	172
5.74.2.11 mergeJson()	172
5.74.2.12 qualifyObject()	173
5.75 ModuleError Class Reference	173
5.75.1 Detailed Description	175
5.76 ModuleErrorException Class Reference	175
5.76.1 Detailed Description	177
5.77 ModuleProvider Class Reference	177
5.77.1 Detailed Description	178
5.77.2 Member Function Documentation	178
5.77.2.1 addModule()	178
5.77.2.2 cacheSymbol()	178
5.77.2.3 clearCache()	179
5.77.2.4 findCached()	179
5.77.2.5 getInstance()	179
5.77.2.6 getModule()	179
5.77.2.7 hasModule()	179
5.78 MulExpression Class Reference	180
5.78.1 Member Function Documentation	181

5.78.1.1 accept()	181
5.78.1.2 opsign()	182
5.79 NeqExpression Class Reference	182
5.79.1 Member Function Documentation	183
5.79.1.1 accept()	183
5.79.1.2 opsign()	184
5.80 node_w_depth_t Struct Reference	184
5.80.1 Detailed Description	184
5.81 NoneExpression Class Reference	184
5.81.1 Detailed Description	186
5.81.2 Member Function Documentation	186
5.81.2.1 accept()	186
5.81.2.2 getTextFormatted()	187
5.82 NotExpression Class Reference	187
5.82.1 Member Function Documentation	188
5.82.1.1 accept()	189
5.83 ObservingVisitor Class Reference	189
5.83.1 Detailed Description	191
5.83.2 Member Function Documentation	191
5.83.2.1 start()	191
5.83.2.2 visitBinaryOperatorDefault()	191
5.84 OfExpression Class Reference	192
5.84.1 Detailed Description	193
5.84.2 Member Function Documentation	193
5.84.2.1 accept()	193
5.84.2.2 areOperandsValid()	193
5.84.2.3 getQuantifier()	193
5.84.2.4 getSet()	194
5.84.2.5 getTextFormatted()	194
5.84.2.6 getType()	194
5.84.2.7 isComplete()	194
5.84.2.8 isValid()	194
5.85 offset_edit_range_t Struct Reference	195
5.85.1 Detailed Description	195
5.85.2 Member Enumeration Documentation	195
5.85.2.1 EditType	195
5.85.3 Member Function Documentation	196
5.85.3.1 len()	196
5.86 offset_range_t Struct Reference	196
5.86.1 Detailed Description	196
5.87 OrExpression Class Reference	197
5.87.1 Member Function Documentation	198

5.87.1.1 accept()	198
5.87.1.2 opsign()	199
5.88 OverflowError Class Reference	199
5.88.1 Detailed Description	200
5.89 OverflowErrorException Class Reference	200
5.89.1 Detailed Description	202
5.90 ParenthesesExpression Class Reference	202
5.90.1 Detailed Description	204
5.90.2 Member Function Documentation	204
5.90.2.1 accept()	204
5.90.2.2 areOperandsValid()	204
5.90.2.3 getTextFormatted()	204
5.90.2.4 getType()	205
5.90.2.5 isComplete()	205
5.90.2.6 isValid()	205
5.91 PercentQuantifierExpression Class Reference	206
5.91.1 Member Function Documentation	207
5.91.1.1 accept()	207
5.91.1.2 areOperandsValid()	208
5.91.1.3 getTextFormatted()	208
5.91.1.4 getType()	208
5.91.1.5 isComplete()	208
5.91.1.6 isValid()	208
5.92 PlainSymbol Class Reference	209
5.92.1 Member Function Documentation	210
5.92.1.1 accept()	210
5.92.1.2 getTextFormatted()	211
5.93 point_t Struct Reference	211
5.93.1 Detailed Description	211
5.94 SymTabDuplId< V >::Pool Class Reference	212
5.94.1 Detailed Description	212
5.94.2 Member Function Documentation	212
5.94.2.1 erase()	213
5.95 Printable Class Reference	213
5.95.1 Detailed Description	214
5.95.2 Member Function Documentation	215
5.95.2.1 getTextFormatted()	215
5.96 QuantifierExpression Class Reference	215
5.96.1 Member Function Documentation	216
5.96.1.1 areOperandsValid()	216
5.96.1.2 getType()	217
5.96.1.3 isComplete()	217

5.96.1.4 isValid()	217
5.97 Range Class Reference	218
5.97.1 Detailed Description	219
5.97.2 Member Function Documentation	219
5.97.2.1 accept()	219
5.97.2.2 areOperandsValid()	219
5.97.2.3 getTextFormatted()	220
5.97.2.4 getType()	220
5.98 range_t Struct Reference	220
5.98.1 Detailed Description	221
5.99 RangeError Class Reference	221
5.99.1 Detailed Description	223
5.100 RangeErrorException Class Reference	224
5.100.1 Detailed Description	225
5.101 RegexpError Class Reference	226
5.101.1 Detailed Description	227
5.102 RegexpErrorException Class Reference	228
5.102.1 Detailed Description	229
5.103 RegexpExpression Class Reference	230
5.103.1 Detailed Description	231
5.103.2 Member Function Documentation	231
5.103.2.1 accept()	232
5.103.2.2 getTextFormatted()	232
5.103.2.3 getType()	232
5.103.2.4 isComplete()	232
5.104 RemainderExpression Class Reference	233
5.104.1 Member Function Documentation	234
5.104.1.1 accept()	235
5.104.1.2 getType()	235
5.104.1.3 isValid()	235
5.104.1.4 opsign()	235
5.105 RightShiftExpression Class Reference	236
5.105.1 Member Function Documentation	237
5.105.1.1 accept()	237
5.105.1.2 opsign()	238
5.106 Rule Class Reference	238
5.106.1 Detailed Description	240
5.106.2 Member Function Documentation	240
5.106.2.1 addMeta()	240
5.106.2.2 addModifier()	240
5.106.2.3 addString()	240
5.106.2.4 addTag()	241

5.106.2.5 addVar()	241
5.106.2.6 getCondition()	241
5.106.2.7 getId()	241
5.106.2.8 getIdOffset()	241
5.106.2.9 getMetas()	241
5.106.2.10 getMetasById()	242
5.106.2.11 getModifiers()	242
5.106.2.12 getStringById()	242
5.106.2.13 getStrings()	242
5.106.2.14 getTags()	242
5.106.2.15 getTextFormatted()	242
5.106.2.16 getVarById()	243
5.106.2.17 getVars()	243
5.106.2.18 hasTag()	243
5.106.2.19 isGlobal()	243
5.106.2.20 isPrivate()	243
5.106.2.21 removeMeta()	243
5.106.2.22 removeModifier()	244
5.106.2.23 removeString()	244
5.106.2.24 removeTag()	244
5.106.2.25 removeVar()	244
5.106.2.26 setCondition()	244
5.106.2.27 setId()	244
5.106.2.28 setIdOffset()	245
5.107 RuleElement Class Reference	245
5.107.1 Detailed Description	246
5.107.2 Member Function Documentation	246
5.107.2.1 getLocalOffset()	246
5.107.2.2 getOffset()	247
5.107.2.3 getParentRule()	247
5.107.2.4 getRange()	247
5.107.2.5 setLocalOffset()	247
5.107.2.6 setOffset()	247
5.107.2.7 setParentRule()	247
5.108 RuleError Class Reference	248
5.108.1 Detailed Description	249
5.109 RuleErrorException Class Reference	249
5.109.1 Detailed Description	251
5.110 RuleModifier Class Reference	251
5.110.1 Detailed Description	252
5.110.2 Member Enumeration Documentation	253
5.110.2.1 Type	253

5.110.3 Member Function Documentation	253
5.110.3.1 getType()	253
5.110.3.2 stringToType()	253
5.110.3.3 typeToString()	253
5.111 RuleModifierContainer Class Reference	254
5.111.1 Detailed Description	254
5.111.2 Member Function Documentation	254
5.111.2.1 add()	254
5.111.2.2 begin()	254
5.111.2.3 empty()	255
5.111.2.4 end()	255
5.111.2.5 erase()	255
5.111.2.6 getGlobal()	255
5.111.2.7 getPrivate()	255
5.111.2.8 has()	255
5.111.2.9 hasGlobal()	256
5.111.2.10 hasPrivate()	256
5.111.2.11 size()	256
5.112 RuleModifierError Class Reference	256
5.112.1 Detailed Description	258
5.113 RuleModifierErrorException Class Reference	259
5.113.1 Detailed Description	260
5.114 RuleReferenceError Class Reference	261
5.114.1 Detailed Description	262
5.115 RuleReferenceErrorException Class Reference	263
5.115.1 Detailed Description	264
5.116 RuleSetExpression Class Reference	265
5.116.1 Detailed Description	266
5.116.2 Member Function Documentation	266
5.116.2.1 accept()	267
5.116.2.2 getTextFormatted()	267
5.117 RuleWildcardExpression Class Reference	268
5.117.1 Member Function Documentation	269
5.117.1.1 accept()	269
5.118 SemanticError Class Reference	270
5.118.1 Detailed Description	271
5.118.2 Member Function Documentation	271
5.118.2.1 isGlobal()	271
5.118.2.2 setGlobal()	271
5.119 SemanticErrorException Class Reference	272
5.119.1 Detailed Description	273
5.120 SetExpression Class Reference	273

5.120.1 Detailed Description	274
5.120.2 Member Function Documentation	274
5.120.2.1 areOperandsValid()	275
5.120.2.2 getElements()	275
5.120.2.3 getType()	275
5.120.2.4 isComplete()	275
5.120.2.5 isValid()	275
5.121 SizeExpression Class Reference	276
5.121.1 Member Function Documentation	277
5.121.1.1 accept()	278
5.121.1.2 getTextFormatted()	278
5.121.1.3 getType()	278
5.121.1.4 isComplete()	278
5.121.1.5 unitToString()	279
5.122 StartsWithExpression Class Reference	279
5.122.1 Member Function Documentation	280
5.122.1.1 accept()	280
5.122.1.2 opsign()	281
5.123 String Class Reference	281
5.123.1 Detailed Description	282
5.123.2 Member Enumeration Documentation	282
5.123.2.1 Type	282
5.123.3 Member Function Documentation	283
5.123.3.1 addModifier()	283
5.123.3.2 getContent()	283
5.123.3.3 getId()	283
5.123.3.4 getModifiers()	283
5.123.3.5 getTextFormatted()	283
5.123.3.6 getType()	284
5.123.3.7 isAnonymous()	284
5.123.3.8 setAnonymous()	284
5.123.3.9 setContent()	284
5.123.3.10 setId()	284
5.123.3.11 typeToString()	285
5.124 string_hash_t Struct Reference	285
5.124.1 Detailed Description	285
5.125 StringCountExpression Class Reference	286
5.125.1 Member Function Documentation	287
5.125.1.1 accept()	287
5.125.1.2 getTextFormatted()	288
5.125.1.3 getType()	288
5.125.1.4 isComplete()	288

5.126 StringError Class Reference	289
5.126.1 Detailed Description	290
5.127 StringErrorException Class Reference	290
5.127.1 Detailed Description	292
5.128 StringExpression Class Reference	292
5.128.1 Detailed Description	293
5.128.2 Member Function Documentation	293
5.128.2.1 accept()	294
5.128.2.2 getTextFormatted()	294
5.128.2.3 getType()	294
5.128.2.4 isComplete()	294
5.129 StringMatchLengthExpression Class Reference	295
5.129.1 Member Function Documentation	296
5.129.1.1 accept()	296
5.129.1.2 getTextFormatted()	297
5.129.1.3 getType()	297
5.130 StringModifier Class Reference	297
5.130.1 Detailed Description	299
5.130.2 Member Function Documentation	299
5.130.2.1 getArg()	299
5.130.2.2 getTextFormatted()	299
5.130.2.3 getType()	299
5.130.2.4 hasAnyArg()	299
5.130.2.5 hasArg()	299
5.130.2.6 setArg()	300
5.130.2.7 stringToType()	300
5.130.2.8 typeToString()	300
5.131 StringModifierContainer Class Reference	300
5.131.1 Detailed Description	301
5.131.2 Member Function Documentation	301
5.131.2.1 add()	301
5.131.2.2 begin()	301
5.131.2.3 empty()	301
5.131.2.4 end()	301
5.131.2.5 erase()	302
5.131.2.6 get()	302
5.131.2.7 has()	302
5.131.2.8 size()	302
5.132 StringModifierError Class Reference	303
5.132.1 Detailed Description	304
5.133 StringModifierErrorException Class Reference	305
5.133.1 Detailed Description	306

5.134 StringOffsetExpression Class Reference	307
5.134.1 Member Function Documentation	308
5.134.1.1 accept()	308
5.134.1.2 getTextFormatted()	309
5.134.1.3 getType()	309
5.135 StringPropertyExpression Class Reference	309
5.135.1 Member Function Documentation	310
5.135.1.1 areOperandsValid()	310
5.135.1.2 isComplete()	311
5.135.1.3 isValid()	311
5.136 StringReferenceError Class Reference	311
5.136.1 Detailed Description	313
5.137 StringReferenceErrorException Class Reference	314
5.137.1 Detailed Description	315
5.138 StringSetExpression Class Reference	316
5.138.1 Detailed Description	317
5.138.2 Member Function Documentation	317
5.138.2.1 accept()	318
5.138.2.2 getTextFormatted()	318
5.139 StringWildcardExpression Class Reference	319
5.139.1 Member Function Documentation	320
5.139.1.1 accept()	320
5.140 StructExpression Class Reference	321
5.140.1 Member Function Documentation	322
5.140.1.1 accept()	323
5.140.1.2 areOperandsValid()	323
5.140.1.3 getTextFormatted()	323
5.140.1.4 getType()	323
5.140.1.5 isComplete()	323
5.140.1.6 isValid()	323
5.141 SubExpression Class Reference	324
5.141.1 Member Function Documentation	325
5.141.1.1 accept()	325
5.141.1.2 opsign()	326
5.142 Symbol Class Reference	326
5.142.1 Detailed Description	328
5.142.2 Member Function Documentation	328
5.142.2.1 areOperandsValid()	328
5.142.2.2 getContext()	328
5.142.2.3 getSymType()	328
5.142.2.4 getTextFormatted()	329
5.142.2.5 getType()	329

5.142.2.6 isComplete()	329
5.142.2.7 isDefined()	329
5.142.2.8 isValid()	329
5.142.2.9 setContext()	330
5.142.2.10 setDefined()	330
5.142.2.11 setSymType()	330
5.142.3 Member Data Documentation	330
5.142.3.1 ctx_	330
5.142.3.2 defined_	331
5.142.3.3 dtype_	331
5.142.3.4 stype_	331
5.143 SymbolError Class Reference	331
5.143.1 Detailed Description	333
5.144 SymbolErrorException Class Reference	334
5.144.1 Detailed Description	335
5.145 SymbolReferenceError Class Reference	336
5.145.1 Detailed Description	337
5.146 SymbolReferenceErrorException Class Reference	338
5.146.1 Detailed Description	339
5.147 SymTab< V > Class Template Reference	339
5.147.1 Detailed Description	340
5.147.2 Member Function Documentation	340
5.147.2.1 clear()	341
5.147.2.2 dataById()	341
5.147.2.3 dataByOffset()	341
5.147.2.4 insert()	341
5.147.2.5 remove() [1/2]	341
5.147.2.6 remove() [2/2]	342
5.147.2.7 revRemoveUntil()	342
5.147.2.8 search()	342
5.147.2.9 update()	342
5.148 SymTabDuplId< V > Class Template Reference	343
5.148.1 Detailed Description	344
5.148.2 Member Function Documentation	344
5.148.2.1 at()	344
5.148.2.2 clear()	344
5.148.2.3 exists()	344
5.148.2.4 getAll()	344
5.148.2.5 insert()	344
5.148.2.6 remove() [1/2]	345
5.148.2.7 remove() [2/2]	345
5.148.2.8 revRemoveUntil()	345

5.148.2.9 update()	345
5.149 SymTabOffset< V > Class Template Reference	346
5.149.1 Detailed Description	346
5.149.2 Member Function Documentation	347
5.149.2.1 exists()	347
5.149.2.2 insert()	347
5.149.2.3 remove()	347
5.149.2.4 revRemoveUntil()	347
5.149.2.5 update() [1/2]	347
5.149.2.6 update() [2/2]	348
5.150 SyntaxError Class Reference	348
5.150.1 Detailed Description	349
5.151 SyntaxErrorException Class Reference	349
5.151.1 Detailed Description	351
5.152 TerminalExpression Class Reference	351
5.152.1 Detailed Description	352
5.152.2 Member Function Documentation	352
5.152.2.1 areOperandsValid()	352
5.152.2.2 isValid()	352
5.153 ThemExpression Class Reference	353
5.153.1 Detailed Description	354
5.153.2 Member Function Documentation	354
5.153.2.1 accept()	354
5.153.2.2 areOperandsValid()	354
5.153.2.3 getTextFormatted()	354
5.153.2.4 getType()	355
5.153.2.5 isComplete()	355
5.153.2.6 isValid()	355
5.154 TopLevelYaraFileElement Class Reference	355
5.154.1 Detailed Description	357
5.154.2 Member Function Documentation	357
5.154.2.1 getPosition()	357
5.154.2.2 getRange()	357
5.154.2.3 hasValidCache()	357
5.154.2.4 invalidateCache()	357
5.154.2.5 setRangeCache()	358
5.155 TSParserAdapter Class Reference	358
5.155.1 Detailed Description	359
5.155.2 Member Function Documentation	359
5.155.2.1 createYaraSource()	359
5.155.2.2 createYaraSourceFromFile()	359
5.155.2.3 parse()	360

5.155.2.4 reparse()	360
5.155.2.5 updateYaraSource()	360
5.155.2.6 updateYaraSourceFromFile()	361
5.156 UnaryArithmeticExpression Class Reference	361
5.156.1 Detailed Description	363
5.156.2 Member Function Documentation	363
5.156.2.1 getType()	363
5.156.2.2 isValid()	364
5.157 UnaryBitwiseExpression Class Reference	364
5.157.1 Detailed Description	366
5.157.2 Member Function Documentation	366
5.157.2.1 getType()	367
5.157.2.2 isValid()	367
5.158 UnaryBoolExpression Class Reference	367
5.158.1 Detailed Description	369
5.158.2 Member Function Documentation	369
5.158.2.1 getType()	369
5.158.2.2 isValid()	370
5.159 UnaryExpression Class Reference	370
5.159.1 Detailed Description	371
5.159.2 Member Function Documentation	371
5.159.2.1 areOperandsValid()	372
5.159.2.2 getTextFormatted()	372
5.159.2.3 isComplete()	372
5.160 UnaryMinusExpression Class Reference	373
5.160.1 Member Function Documentation	374
5.160.1.1 accept()	375
5.161 UnexpectedTokenError Class Reference	375
5.161.1 Detailed Description	376
5.162 UnexpectedTokenErrorException Class Reference	376
5.162.1 Detailed Description	378
5.163 var_def_t Struct Reference	378
5.163.1 Detailed Description	379
5.164 Variable Class Reference	379
5.164.1 Detailed Description	380
5.164.2 Member Function Documentation	380
5.164.2.1 getId()	380
5.164.2.2 setId()	380
5.165 VariableError Class Reference	380
5.165.1 Detailed Description	382
5.166 VariableErrorException Class Reference	382
5.166.1 Detailed Description	384

5.167 VariableExpression Class Reference	385
5.167.1 Member Function Documentation	386
5.167.1.1 accept()	386
5.168 VarListExpression Class Reference	387
5.168.1 Detailed Description	388
5.168.2 Member Function Documentation	388
5.168.2.1 accept()	388
5.168.2.2 areOperandsValid()	388
5.168.2.3 getTextFormatted()	389
5.168.2.4 getType()	389
5.168.2.5 isComplete()	389
5.168.2.6 isValid()	389
5.169 Visitor Class Reference	390
5.169.1 Detailed Description	391
5.170 YaraErrorException Class Reference	392
5.170.1 Detailed Description	393
5.171 YaraFile Class Reference	393
5.171.1 Detailed Description	396
5.171.2 Member Function Documentation	396
5.171.2.1 addComment()	396
5.171.2.2 addEdit()	396
5.171.2.3 addError()	396
5.171.2.4 addFileInclude()	397
5.171.2.5 addImport()	397
5.171.2.6 addMissingSymbol()	397
5.171.2.7 addRule()	397
5.171.2.8 addSemanticError()	397
5.171.2.9 addSyntaxError()	398
5.171.2.10 addWrong()	398
5.171.2.11 bind() [1/2]	398
5.171.2.12 bind() [2/2]	398
5.171.2.13 checkDuplicatedRules()	398
5.171.2.14 checkMissing()	399
5.171.2.15 checkRange()	399
5.171.2.16 clearAll()	399
5.171.2.17 clearComments()	399
5.171.2.18 clearEdits()	399
5.171.2.19 clearLocalFileIncludes()	399
5.171.2.20 clearLocalImports()	399
5.171.2.21 clearLocalRules()	400
5.171.2.22 clearSemanticErrors()	400
5.171.2.23 clearSyntaxErrors()	400

5.171.2.24 clearToBeChecked()	400
5.171.2.25 clearWrong()	400
5.171.2.26 createRuleError()	400
5.171.2.27 ctx()	400
5.171.2.28 deleteBindings()	401
5.171.2.29 dumpSyntaxErrors()	401
5.171.2.30 edit() [1/2]	401
5.171.2.31 edit() [2/2]	401
5.171.2.32 getCommentByIndex()	402
5.171.2.33 getComments()	402
5.171.2.34 getEdits()	402
5.171.2.35 getFileIncludes()	402
5.171.2.36 getImport() [1/4]	402
5.171.2.37 getImport() [2/4]	403
5.171.2.38 getImport() [3/4]	403
5.171.2.39 getImport() [4/4]	403
5.171.2.40 getImports()	404
5.171.2.41 getLocalFileInclude()	404
5.171.2.42 getLocalFileIncludes()	404
5.171.2.43 getLocalImport()	404
5.171.2.44 getLocalImports()	404
5.171.2.45 getLocalRule()	404
5.171.2.46 getLocalRuleByRow()	404
5.171.2.47 getLocalRules()	405
5.171.2.48 getName()	405
5.171.2.49 getRule() [1/3]	405
5.171.2.50 getRule() [2/3]	405
5.171.2.51 getRule() [3/3]	406
5.171.2.52 getRules() [1/2]	406
5.171.2.53 getRules() [2/2]	406
5.171.2.54 getSemanticErrors()	406
5.171.2.55 getSyntaxErrors()	406
5.171.2.56 getTextFormatted()	407
5.171.2.57 hasFastEdit()	407
5.171.2.58 hasRule() [1/2]	407
5.171.2.59 hasRule() [2/2]	407
5.171.2.60 isActive()	407
5.171.2.61 isIsolated()	407
5.171.2.62 modify()	408
5.171.2.63 mustBeReparsed()	408
5.171.2.64 notifyAddedSymbol()	408
5.171.2.65 notifyChange()	408

5.171.2.66 removeComments()	408
5.171.2.67 removeElementsInRange()	408
5.171.2.68 removeGlobalErrors()	409
5.171.2.69 removeLocalFileIncludes()	409
5.171.2.70 removeLocalImports()	409
5.171.2.71 removeLocalRules()	409
5.171.2.72 removeMissingSymbol()	409
5.171.2.73 removeSemanticErrors()	409
5.171.2.74 removeSyntaxErrors()	410
5.171.2.75 ruleLookUp()	410
5.171.2.76 setActivity()	410
5.171.2.77 setErrorMode()	410
5.171.2.78 setIsolated()	410
5.171.2.79 setName()	411
5.171.2.80 shiftElements()	411
5.171.2.81 undo()	411
5.171.2.82 updateComments()	411
5.171.2.83 updateLocalFileIncludes()	411
5.171.2.84 updateLocalImports()	411
5.171.2.85 updateLocalRules()	412
5.171.2.86 updateSyntaxErrors()	412
5.171.2.87 wasEdited()	412
5.172 YaraFileElement Class Reference	412
5.172.1 Detailed Description	414
5.172.2 Member Function Documentation	414
5.172.2.1 getLen()	414
5.172.2.2 getOffset()	414
5.172.2.3 getParentFile()	414
5.172.2.4 getPosition()	415
5.172.2.5 getRange()	415
5.172.2.6 isDeletionRequested()	415
5.172.2.7 isFixed()	415
5.172.2.8 requestDeletion()	415
5.172.2.9 setLen()	415
5.172.2.10 setOffset()	415
5.172.2.11 setParent()	416
5.172.2.12 setParentFile()	416
5.173 YaraFileElementBindable Class Reference	416
5.173.1 Detailed Description	417
5.173.2 Member Function Documentation	417
5.173.2.1 bind()	417
5.173.2.2 bind_to()	417

5.174 Yaramod Class Reference	417
5.174.1 Detailed Description	418
5.174.2 Member Function Documentation	418
5.174.2.1 config()	418
5.174.2.2 loadModules()	418
5.174.2.3 loadModulesFromEnvironment()	418
5.174.2.4 loadModulesFromPath()	419
5.174.2.5 parseFile() [1/2]	419
5.174.2.6 parseFile() [2/2]	419
5.174.2.7 parseString() [1/2]	420
5.174.2.8 parseString() [2/2]	420
5.175 YaramodConfig Class Reference	421
5.175.1 Detailed Description	421
5.175.2 Member Enumeration Documentation	421
5.175.2.1 ErrorMode	421
5.175.2.2 ParsingMode	422
5.175.3 Constructor & Destructor Documentation	422
5.175.3.1 YaramodConfig()	422
5.175.4 Member Function Documentation	422
5.175.4.1 addBoolVar()	422
5.175.4.2 addVar()	423
5.175.4.3 getBaseDir()	423
5.175.4.4 getErrorMode()	423
5.175.4.5 getModuleDir()	423
5.175.4.6 getParsingMode()	423
5.175.4.7 getVar()	423
5.175.4.8 removeVar()	424
5.175.4.9 setBaseDir()	424
5.175.4.10 setErrorMode()	424
5.175.4.11 setModuleDir()	424
5.175.4.12 setParsingMode()	425
5.176 YaramodErrorException Class Reference	425
5.176.1 Detailed Description	426
5.177 YaraSource Class Reference	426
5.177.1 Detailed Description	428
5.177.2 Member Function Documentation	428
5.177.2.1 addFile()	428
5.177.2.2 bind()	428
5.177.2.3 clearInactiveFiles()	428
5.177.2.4 deactivateFiles()	428
5.177.2.5 deleteBindings()	428
5.177.2.6 dumpBindChildren()	429

5.177.2.7 dumpBindParents()	429
5.177.2.8 edit() [1/4]	429
5.177.2.9 edit() [2/4]	429
5.177.2.10 edit() [3/4]	430
5.177.2.11 edit() [4/4]	430
5.177.2.12 getAllRules()	431
5.177.2.13 getAllSemanticErrors()	431
5.177.2.14 getAllSyntaxErrors()	431
5.177.2.15 getBaseDir()	432
5.177.2.16 getEntryFile()	432
5.177.2.17 getFile()	432
5.177.2.18 getFiles()	432
5.177.2.19 getImport()	432
5.177.2.20 getRule()	433
5.177.2.21 getTextFormatted()	433
5.177.2.22 hasFile()	433
5.177.2.23 moveBindings()	433
5.177.2.24 notifyChange()	434
5.177.2.25 removeFile()	434
5.177.2.26 renameFile()	434
5.177.2.27 setEntryFile()	434
5.177.2.28 undo()	434
6 File Documentation	435
6.1 /home/vojtech.dvorak1/Documents/yaramod-v4/src/common.cpp File Reference	435
6.1.1 Detailed Description	436
6.1.2 Function Documentation	436
6.1.2.1 normalizeFSPath()	436
6.1.2.2 offsetToPoint()	436
6.1.2.3 pointToOffset()	436
6.2 /home/vojtech.dvorak1/Documents/yaramod-v4/src/error.cpp File Reference	437
6.2.1 Detailed Description	437
6.3 /home/vojtech.dvorak1/Documents/yaramod-v4/src/file_context.cpp File Reference	437
6.3.1 Detailed Description	438
6.4 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/common.h File Reference	438
6.4.1 Detailed Description	440
6.4.2 Function Documentation	440
6.4.2.1 normalizeFSPath()	440
6.4.2.2 offsetToPoint()	440
6.4.2.3 pointToOffset()	440
6.5 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h File Reference	441
6.5.1 Detailed Description	443

6.6 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error_collector.h File Reference	444
6.6.1 Detailed Description	444
6.7 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h File Reference	445
6.8 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression_types.h File Reference	448
6.8.1 Detailed Description	448
6.9 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/file_context.h File Reference	448
6.9.1 Detailed Description	449
6.10 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/forward.h File Reference	449
6.10.1 Detailed Description	449
6.11 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/language.h File Reference	450
6.11.1 Detailed Description	450
6.11.2 Function Documentation	451
6.11.2.1 escapeSequenceToChar()	451
6.11.2.2 getBuiltinSymbol()	451
6.11.2.3 isValidEscapeSequence()	451
6.11.2.4 isYaraKeyword()	452
6.11.2.5 tree_sitter_yara()	452
6.12 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/literal.h File Reference	452
6.12.1 Detailed Description	453
6.13 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/meta.h File Reference	453
6.13.1 Detailed Description	454
6.14 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/module.h File Reference	454
6.14.1 Detailed Description	455
6.15 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/observing_visitor.h File Reference	455
6.15.1 Detailed Description	457
6.16 /home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/visitor/observing_visitor.h File Reference	457
6.16.1 Detailed Description	457
6.17 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/other.h File Reference	458
6.17.1 Detailed Description	458
6.18 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/parser_adapter.h File Reference	459
6.18.1 Detailed Description	460
6.18.2 Macro Definition Documentation	460
6.18.2.1 INITIAL_NODE_STACK_SIZE	460
6.19 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule.h File Reference	460
6.19.1 Detailed Description	461
6.20 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule_element.h File Reference	461
6.20.1 Detailed Description	462
6.21 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/string.h File Reference	462
6.21.1 Detailed Description	463
6.21.2 Macro Definition Documentation	463
6.21.2.1 BASE64_ALPHA_LEN	463
6.21.2.2 REGEXP_MAXIMUM_QUANTIFIER_NUM	464

6.21.2.3 XOR_UPPER_RANGE_BOUND	464
6.22 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/symtab.h File Reference	464
6.22.1 Detailed Description	465
6.23 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/types.h File Reference	465
6.23.1 Detailed Description	466
6.24 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/variable.h File Reference	467
6.24.1 Detailed Description	468
6.25 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/visitor.h File Reference	468
6.25.1 Detailed Description	469
6.26 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file.h File Reference	469
6.26.1 Detailed Description	470
6.27 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file_element.h File Reference	470
6.27.1 Detailed Description	471
6.28 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_source.h File Reference	471
6.28.1 Detailed Description	472
6.29 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yaramod.h File Reference	472
6.29.1 Detailed Description	473
6.30 /home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/yaramod.h File Reference	473
6.30.1 Detailed Description	473
6.31 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yaramod_config.h File Reference	474
6.31.1 Detailed Description	474
6.32 /home/vojtech.dvorak1/Documents/yaramod-v4/src/language.cpp File Reference	475
6.32.1 Detailed Description	475
6.32.2 Function Documentation	475
6.32.2.1 escapeSequenceToChar()	475
6.32.2.2 getBuiltinSymbol()	476
6.32.2.3 isValidEscapeSequence()	476
6.32.2.4 isYaraKeyword()	476
6.33 /home/vojtech.dvorak1/Documents/yaramod-v4/src/literal.cpp File Reference	477
6.33.1 Detailed Description	477
6.34 /home/vojtech.dvorak1/Documents/yaramod-v4/src/module.cpp File Reference	477
6.34.1 Detailed Description	478
6.35 /home/vojtech.dvorak1/Documents/yaramod-v4/src/other.cpp File Reference	478
6.35.1 Detailed Description	479
6.36 /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter.cpp File Reference	479
6.36.1 Detailed Description	479
6.37 /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter_expression.cpp File Reference	480
6.37.1 Detailed Description	480
6.38 /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter_rule.cpp File Reference	480
6.38.1 Detailed Description	481
6.39 /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp File Reference	481

6.39.1 Detailed Description	481
6.40 /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/meta.cpp File Reference	482
6.40.1 Detailed Description	482
6.41 /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/rule_element.cpp File Reference	482
6.41.1 Detailed Description	483
6.42 /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/string.cpp File Reference	483
6.42.1 Detailed Description	483
6.43 /home/vojtech.dvorak1/Documents/yaramod-v4/src/variable.cpp File Reference	484
6.43.1 Detailed Description	484
6.44 /home/vojtech.dvorak1/Documents/yaramod-v4/src/visitor/observing_visitor.cpp File Reference	484
6.44.1 Detailed Description	485
6.45 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file.cpp File Reference	485
6.45.1 Detailed Description	486
6.45.2 Function Documentation	486
6.45.2.1 updateModifiedRanges()	486
6.46 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file_element.cpp File Reference	487
6.46.1 Detailed Description	487
6.47 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_source.cpp File Reference	487
6.47.1 Detailed Description	488
6.48 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod.cpp File Reference	488
6.48.1 Detailed Description	488
6.49 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_config.cpp File Reference	489
6.49.1 Detailed Description	489
Index	491

Chapter 1

Documentation main page

This library provides interface for parsing and incremental reparsing of source file in YARA language.

The results of parsing (which includes syntax analysis as well as semantic analysis) can be accessed by high level semantic interface. The main advantage over the yaramod-v3 is the possibility of incremental reparsing - edits of YARA source can be reported continuously and then the mentioned high level representation can be easily updated. This approach is much more faster than parsing from the scratch.

See also

What is the YARA language? <https://yara.readthedocs.io/en/stable/>

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

FileContext::cache_entry_t	53
FileContext::cache_t	54
edit_t	64
std::enable_shared_from_this	
YaraFile	393
ErrorCollector< E >	73
ErrorCollector< SemanticError >	73
ErrorCollector< SyntaxError >	73
std::exception	
ErrorException	76
InternalErrorException	136
YaraErrorException	392
SemanticErrorException	272
ExpressionErrorException	83
FunctionCallErrorException	103
IndexErrorException	131
RangeErrorException	224
RuleReferenceErrorException	263
StringReferenceErrorException	314
SymbolErrorException	334
SymbolReferenceErrorException	338
IncludeErrorException	126
ModuleErrorException	175
OverflowErrorException	200
RuleErrorException	249
RuleModifierErrorException	259
StringErrorException	290
HexStrErrorException	115
RegexpErrorException	228
StringModifierErrorException	305
VariableErrorException	382
SyntaxErrorException	349
MissingTokenErrorException	168
UnexpectedTokenErrorException	376

YaramodErrorException	425
FileContext	89
IParserAdapter	139
TSParserAdapter	358
RuleModifierContainer::iterator	144
StringModifierContainer::iterator	145
SymTab< V >::iterator	145
SymTabDuplId< V >::iterator	146
SymTabDuplId< V >::Pool::iterator	147
SymTabOffset< V >::iterator	148
Module	169
ModuleProvider	177
node_w_depth_t	184
offset_edit_range_t	195
offset_range_t	196
point_t	211
SymTabDuplId< V >::Pool	212
Printable	213
Comment	55
Expression	77
BinaryExpression	37
AtExpression	28
BinaryArithmeticExpression	31
AddExpression	17
DivExpression	62
MulExpression	180
RemainderExpression	233
SubExpression	324
BinaryBitwiseExpression	33
BitwiseAndExpression	45
BitwiseOrExpression	49
BitwiseXorExpression	51
LeftShiftExpression	149
RightShiftExpression	236
BinaryBoolExpression	35
AndExpression	21
OrExpression	197
BinaryRelationalExpression	41
EqExpression	70
GtExpression	111
GteExpression	109
LtExpression	159
LteExpression	157
NeqExpression	182
BinaryStrRelationalExpression	43
ContainsExpression	57
EndsWithExpression	65
IContainsExpression	117
IEndsWithExpression	119
IEqExpression	121
IStartsWithExpression	142
StartsWithExpression	279
MatchesExpression	161
Enum	67
ForExpression	94
ForIntExpression	97
InExpression	133

OfExpression	192
ParenthesesExpression	202
QuantifierExpression	215
AllExpression	19
AnyExpression	23
NoneExpression	184
PercentQuantifierExpression	206
Range	218
SetExpression	273
ExpressionSetExpression	85
RuleSetExpression	265
StringSetExpression	316
StringPropertyExpression	309
StringMatchLengthExpression	295
StringOffsetExpression	307
Symbol	326
ArrayExpression	25
FunctionCallExpression	105
PlainSymbol	209
RuleWildcardExpression	268
VariableExpression	385
StructExpression	321
TerminalExpression	351
LiteralExpression	153
RegexExpression	230
SizeExpression	276
StringCountExpression	286
StringExpression	292
StringWildcardExpression	319
ThemExpression	353
UnaryExpression	370
DefinedExpression	59
UnaryArithmeticExpression	361
UnaryMinusExpression	373
UnaryBitwiseExpression	364
BitwiseNotExpression	47
UnaryBoolExpression	367
NotExpression	187
VarListExpression	387
FileInclude	92
Import	123
IntVariable	137
Literal	151
Meta	164
Rule	238
String	281
StringModifier	297
YaraFile	393
YaraSource	426
range_t	220
RuleModifierContainer	254
string_hash_t	285
StringModifierContainer	300
SymTab< V >	339
SymTab< FileInclude >	339
SymTab< IntVariable >	339
SymTab< Rule >	339

SymTab< String >	339
SymTabDuplId< V >	343
SymTabDuplId< Import >	343
SymTabDuplId< Meta >	343
SymTabOffset< V >	346
SymTabOffset< Comment >	346
SymTabOffset< E >	346
SymTabOffset< TopLevelYaraFileElement >	346
var_def_t	378
Variable	379
ExtVariable	87
IntVariable	137
Visitor	390
ObservingVisitor	189
YaraFileElement	412
RuleElement	245
Expression	77
IntVariable	137
Meta	164
String	281
StringModifier	297
RuleModifier	251
YaraFileElementBindable	416
TopLevelYaraFileElement	355
Comment	55
Error	72
SemanticError	270
ExpressionError	82
FunctionCallError	100
IndexError	128
RangeError	221
RuleReferenceError	261
StringReferenceError	311
SymbolError	331
SymbolReferenceError	336
IncludeError	125
ModuleError	173
OverflowError	199
RuleError	248
RuleModifierError	256
StringError	289
HexStrError	113
RegexError	226
StringModifierError	303
VariableError	380
SyntaxError	348
MissingTokenError	166
UnexpectedTokenError	375
FileInclude	92
Import	123
Rule	238
Yaramod	417
YaramodConfig	421

Chapter 3

Class Index

3.1 Class List

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

AddExpression	17
AllExpression	
Represents 'all' keyword	19
AndExpression	21
AnyExpression	
Represents 'any' keyword	23
ArrayExpression	25
AtExpression	28
BinaryArithmeticExpression	
Groups all binary arithmetics operators for more generic type checking	31
BinaryBitwiseExpression	
Groups all binary bitwise operators for more generic type checking	33
BinaryBoolExpression	
Groups all binary bitwise operators for more generic type checking	35
BinaryExpression	
Class, that represents binary operators	37
BinaryRelationalExpression	
Groups all binary bitwise operators for more generic type checking	41
BinaryStrRelationalExpression	
Groups all binary bitwise operators for more generic type checking	43
BitwiseAndExpression	45
BitwiseNotExpression	47
BitwiseOrExpression	49
BitwiseXorExpression	51
FileContext::cache_entry_t	
Entry in stored in cache	53
FileContext::cache_t	
Cache of FileContext for offset_t <-> point_t conversion	54
Comment	
Class representing comments (all valid types in YARA language)	55
ContainsExpression	57
DefinedExpression	59
DivExpression	62
edit_t	
Represents textual edit of YaraFile	64

EndsWithExpression	65
Enum	67
EqExpression	70
Error	
Class, that represents errors in analyzed source code. The main difference between Error↔ Exceptions and Errors is in usage - Errors are stored in some vector/list or in something and ErrorException are thrown	72
ErrorCollector< E >	
Class responsible for collecting all types of errors. It is basically a map of vectors. Vectors in map are indexed by the offset where errors occurred in YaraFile	73
ErrorException	
Base class for exceptions	76
Expression	
Represents expression in condition and in definition of internal variables	77
ExpressionError	
Base class for errors of expression (e. g. type mismatch)	82
ExpressionErrorException	
Base class for exception, that represents error, that occurred in expression	83
ExpressionSetExpression	
Represents set of expressions	85
ExtVariable	
Class that represents external variables	87
FileContext	
Wrapper for all necessary data to perform incremental reparsing (they are concrete enough)	89
FileInclude	
Class representing "include <path>" structure in file	92
ForExpression	
Represents 'for' operator that have rule set as iterable set	94
ForIntExpression	
Represents 'for' operator with defined variables for inner expression and with any iterable	97
FunctionCallError	
Bad function call (e. g. bad types or bad count of arguments)	100
FunctionCallErrorException	
Exception for errors, that occur in function call (e. g. bad types of arguments)	103
FunctionCallExpression	105
GteExpression	109
GtExpression	111
HexStrError	
Error in hexadecimal string (e. g. bad jump boundaries)	113
HexStrErrorException	
Exception for semantic errors of hexadecimal string (e. g. bad jump interval)	115
IContainsExpression	117
IEndsWithExpression	119
IEqExpression	121
Import	
Class representing "import <module>" structure in file	123
IncludeError	
Bad file include (e. g. included file does not exists)	125
IncludeErrorException	
Exception for bad file include (e. g. file is included twice)	126
IndexError	
Error of index in array access expression	128
IndexErrorException	
Exception for bad type of index expression in array access expression	131
InExpression	
Class representing 'in' operator	133

InternalErrorException	Exception for unexpected errors, that are probably not caused by user (nonrecoverable errors of internal parser etc.)	136
IntVariable	Class, that represents internal variables	137
IParserAdapter	Interface to which will be connected internal parser by concrete ParserAdapter (the "output socket" of the adapter)	139
IStartsWithExpression	142
RuleModifierContainer::iterator	Iterator for RuleModifierContainer	144
StringModifierContainer::iterator	Iterator for StringModifierContainer	145
SymTab< V >::iterator	SymTab iterator (just a wrapper above OffsetMap::const_iterator)	145
SymTabDuplId< V >::iterator	SymTabDuplId iterator	146
SymTabDuplId< V >::Pool::iterator	Pool iterator	147
SymTabOffset< V >::iterator	SymTabOffset iterator	148
LeftShiftExpression	149
Literal	Object that represents literal, that may appear inside a rule definition	151
LiteralExpression	Represents Literal in expression. Possible terminal node of expression tree (it does not have any other Expression operands)	153
LteExpression	157
LtExpression	159
MatchesExpression	161
Meta	Represents meta information of rule	164
MissingTokenError	Class for syntax error caused missing token	166
MissingTokenErrorException	Exception representing missing token in parsed source code	168
Module	Class representing module (its description)	169
ModuleError	Error that occurred in module import	173
ModuleErrorException	Exception for errors, that occurred in module import 175	
ModuleProvider	Singleton class that provides access to loaded modules. Modules are loaded, when Yaramod class is instantiated	177
MulExpression	180
NeqExpression	182
node_w_depth_t	184
NoneExpression	Represents 'none' keyword	184
NotExpression	187
ObservingVisitor	Concrete Visitor class for observing expression tree	189
OfExpression	Represents all variants of 'of' operator in YARA	192
offset_edit_range_t	Structure for lossless storing data about atomic edits in unified form	195

offset_range_t	
Offset range for specifying the part of file, that was for example modified, deleted etc	196
OrExpression	197
OverflowError	
Overflow error (e. g. integer overflow)	199
OverflowErrorException	
Exception for overflows	200
ParenthesesExpression	
Base class for parentheses () in expressions	202
PercentQuantifierExpression	206
PlainSymbol	209
point_t	
Type for specifying position in input string	211
SymTabDuplId< V >::Pool	
Pool that owns all elements with the same id	212
Printable	
Base class for types, that can be serialized	213
QuantifierExpression	215
Range	
Class, that represents integer range (for example "(1..200)")	218
range_t	
Range in input document specified by start point and end point	220
RangeError	
Bad range (e. g. left bound is greater than right bound)	221
RangeErrorException	
Exception for errors, that occur in range expression (e. g. higher left bound than right bound)	224
RegexpError	
Regular expression error (e. g. bad repeat interval)	226
RegexpErrorException	
Exception of semantic error of regular expression (e. g. bad repeat interval)	228
RegexpExpression	
Represents regular expression literal in expressions	230
RemainderExpression	233
RightShiftExpression	236
Rule	
Class that represents yara rule	238
RuleElement	
Base class for all symbols that are valid in rule context	245
RuleError	
Error, that occurred in error definition (e. g. collision of rule identifiers)	248
RuleErrorException	
Exception for errors occurred in rule definition (e. g. collision of rule identifier)	249
RuleModifier	
Class that represents rule modifiers	251
RuleModifierContainer	254
RuleModifierError	
Error of rule modifier (e. g. duplicated rule modifier)	256
RuleModifierErrorException	
Exception for errors of rule modifier (e. g. duplicated rule modifier)	259
RuleReferenceError	
Bad rule reference (referenced rule does not exists)	261
RuleReferenceErrorException	
Exception for bad rule reference (e. g. rule does not exists)	263
RuleSetExpression	
Represents set of rules	265
RuleWildcardExpression	268
SemanticError	
Base class for semantic errors	270

SemanticErrorException	Exception representing semantic errors in parsed source code	272
SetExpression	Base class for all set expressions	273
SizeExpression	276
StartsWithExpression	279
String	Represents yara string	281
string_hash_t	Hash struct for SymTab , provides heterogeneous comparison lookup	285
StringCountExpression	286
StringError	Error in string definition (e. g. collision of string identifiers)	289
StringErrorException	Exception for errors, that occurred in string definition (e. g. collision of string identifier)	290
StringExpression	Represents string identifier in expression (\$<string_id>)	292
StringMatchLengthExpression	295
StringModifier	Represents modifier of yara string	297
StringModifierContainer	Iterable container for string modifier	300
StringModifierError	Error of string modifier (e. g. invalid type of string modifier for specific string)	303
StringModifierErrorException	Exception error of string modifier (e. g, invalid combination of string type and modifier)	305
StringOffsetExpression	307
StringPropertyExpression	309
StringReferenceError	Bad string reference (referenced string does not exists)	311
StringReferenceErrorException	Exception for bad string reference (e. g. string does not exists)	314
StringSetExpression	Represents set of strings	316
StringWildcardExpression	319
StructExpression	321
SubExpression	324
Symbol	Base class for all symbols in expressions (structures, arrays, variables function calls...)	326
SymbolError	Bad usage of symbol	331
SymbolErrorException	Exception for bad usage of symbol (e. g. symbol is used as structure, but it is value)	334
SymbolReferenceError	Bad symbol reference (e. g. reference to undefined symbol)	336
SymbolReferenceErrorException	Exception for bad symbol reference (e. g. symbol does not exists)	338
SymTab< V >	Tab for symbols with double indexing	339
SymTabDuplId< V >	Almost same type of symbol table as the SymTab , but in this case entries are allowed to have duplicated ids	343
SymTabOffset< V >	Symbol tab for symbols that are indexed only by offsets (which can be duplicated)	346
SyntaxError	Base class for syntax errors	348
SyntaxErrorException	Exception representing syntax error in parsed source code	349

TerminalExpression	
Base class for expression, that have not any subexpressions (operands)	351
ThemExpression	
Represents 'them' keyword	353
TopLevelYaraFileElement	
Class for top level yara file elements	355
TSParserAdapter	
Class responsible for parsing of input string and converting TS structure to high level representation	358
UnaryArithmeticExpression	
Base class for all arithmetic operators (expressions), which have one operand	361
UnaryBitwiseExpression	
Base class for bitwise unary operators (expressions), that have one operand. For example bitwise not (~) operator.	364
UnaryBoolExpression	
Base class for binary operators, that need boolean operands. For example and, or	367
UnaryExpression	
Base class for unary operators (such as unary minus, negation...)	370
UnaryMinusExpression	373
UnexpectedTokenError	
Class for syntax error caused by unexpected token	375
UnexpectedTokenErrorException	
Exception representing syntax error in parsed source code	376
var_def_t	
Structure, that hold all information about in-expression defined variable	378
Variable	
Base class for variables	379
VariableError	
Error of variable definition	380
VariableErrorException	
Exception for errors in variable definition	382
VariableExpression	385
VarListExpression	
Represents list of variables, that are defined in 'for' expression for usage inside internal expression of 'for'	387
Visitor	
Virtual base class for concrete visitor classes	390
YaraErrorException	
Base Exception class for errors located in input YARA code	392
YaraFile	
Represents on continuous sequence of yara rules, file includes and imports	393
YaraFileElement	
Class representing syntax structures of YaraFile (such as Rules, Imports, Includes - everything which has some characters in the source string)	412
YaraFileElementBindable	
Represents of elements of YaraFile , that can have some semantic binding with other elements	416
Yaramod	
The class, that controls parsing of Yara rulesets	417
YaramodConfig	
Objects of this class hold the user defined configuration of parser, that can be changed in runtime	421
YaramodErrorException	
Exception for other errors, that can occur during parsing	425
YaraSource	
Class representing the set of files that are connected by includes	426

Chapter 4

File Index

4.1 File List

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

/home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/yaramod.h	
Include file of yaramod-v4, that contains library interface for parsing of yara rulesets	473
/home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/visitor/observing_visitor.h	
Include file of expression observing visitor, interface, that can be use to inspection of YARA conditions or values of internal variables (Avast feature)	457
/home/vojtech.dvorak1/Documents/yaramod-v4/src/common.cpp	
Implementation of auxiliary functions and methods, that are used across the project	435
/home/vojtech.dvorak1/Documents/yaramod-v4/src/error.cpp	
Implementation of members of error and exception classes	437
/home/vojtech.dvorak1/Documents/yaramod-v4/src/file_context.cpp	
Contains implementation of FileContext methods	437
/home/vojtech.dvorak1/Documents/yaramod-v4/src/language.cpp	
Implementation of functions, that are associated with YARA	475
/home/vojtech.dvorak1/Documents/yaramod-v4/src/literal.cpp	
Implementation class Literal methods	477
/home/vojtech.dvorak1/Documents/yaramod-v4/src/module.cpp	
Implementation of members of Module and ModuleProvider	477
/home/vojtech.dvorak1/Documents/yaramod-v4/src/other.cpp	
Implementation of methods that belongs to "other" elements of YaraFile (includes, imports and comments)	478
/home/vojtech.dvorak1/Documents/yaramod-v4/src/variable.cpp	
Implementation of Variable , IntVariable , ExtVariable methods	484
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_source.cpp	
Implementation of YaraSource methods	487
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/common.h	
Auxiliary functions and classes, that can be used across the project	438
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h	
Contains declaration of error and exception hierarchies	441
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error_collector.h	
Contains template class, that is responsible for collecting errors (and throwing error exceptions)	444
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h	
Contains forward declaration of expression types. It is used to reduce cyclic dependencies of classes that use expressions	445
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression_types.h	
Contains forward declaration of expression types. It is used to reduce cyclic dependencies of classes that use expressions	448
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/file_context.h	
Contains declaration of FileContext class	448

/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ forward.h	449
Contains forward declarations of classes and structures	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ language.h	450
Declarations of functions and constants, that are associated with YARA language (table of key-words etc...)	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ literal.h	452
Contains declaration of Literal class, that represents literals in YaraFile	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ meta.h	453
Contains declaration of Meta class	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ module.h	454
Classes for handling modules (their description in JSON) and their parsing	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ observing_visitor.h	455
Contains declaration of ObservingVisitor class	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ other.h	458
Classes, that represents less complex syntax structures (such as includes, imports, comments...)	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ parser_adapter.h	459
Header file containing abstract IParserAdapter class and concrete TSParserAdapter class	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ rule.h	460
Contains declaration of class, that represents yara rule	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ rule_element.h	461
Contains declaration of RuleElement class	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ string.h	462
Declaration of classes String , StringModifier and StringModifierContainer	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ syntab.h	464
Contains template classes for creating symbol tables	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ types.h	465
Declaration of structures and simple data types, that are used across the project	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ variable.h	467
Declaration of Variable class and its two specializations IntVariable (for internal variables, Avast feature of YARA) and ExtVariables (for external variables)	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ visitor.h	468
Contains declaration of Visitor class	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ yara_file.h	469
Contains YaraFile class, that represents one continuous input in YARA language	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ yara_file_element.h	470
Contains class, that represents syntax structures of YaraFile	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ yara_source.h	471
Contains YaraSource class, that representing the set of yara inputs (string or file with its dependencies)	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ yaramod.h	472
Header file for class representing the parser of the library	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/ yaramod_config.h	474
/home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/ parser_adapter.cpp	479
Implementation of TSParserAdapter class members	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/ parser_adapter_expression.cpp	480
Definitions of methods, that are responsible for converting concrete syntax tree returned by tree-sitter to high level representation	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/ parser_adapter_rule.cpp	480
Implementation of TSParserAdapter members, that are responsible for converting TSNode with rules and rule elements to high level representation	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/ expression.cpp	481
Implementation of Expression class methods	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/ meta.cpp	482
Contains implementation of methods of Meta	
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/ rule_element.cpp	482
Implementation of RuleElement methods	

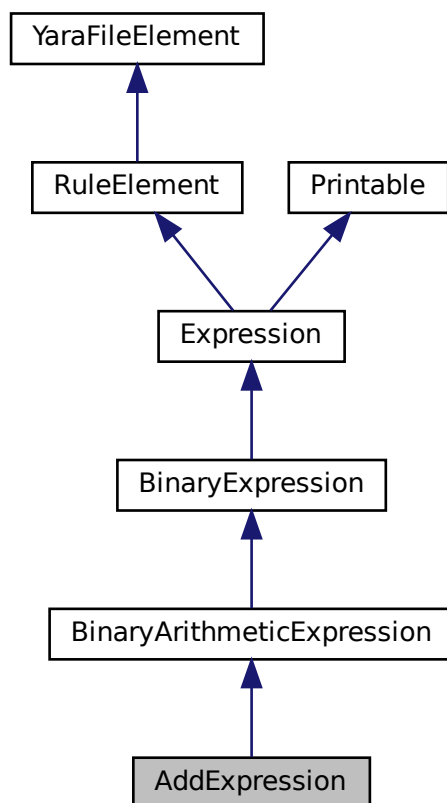
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/string.cpp	
Implementation of String and StringModifier methods	483
/home/vojtech.dvorak1/Documents/yaramod-v4/src/visitor/observing_visitor.cpp	
Contains default implementation of ObservingVisitor methods	484
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file.cpp	
Implementation of YaraFile methods	485
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file_element.cpp	
Implementation of YaraFileElement methods	487
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod.cpp	
Implementation of Yaramod methods	488
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_config.cpp	
Contains implementation of methods of YaraModConfig	489

Chapter 5

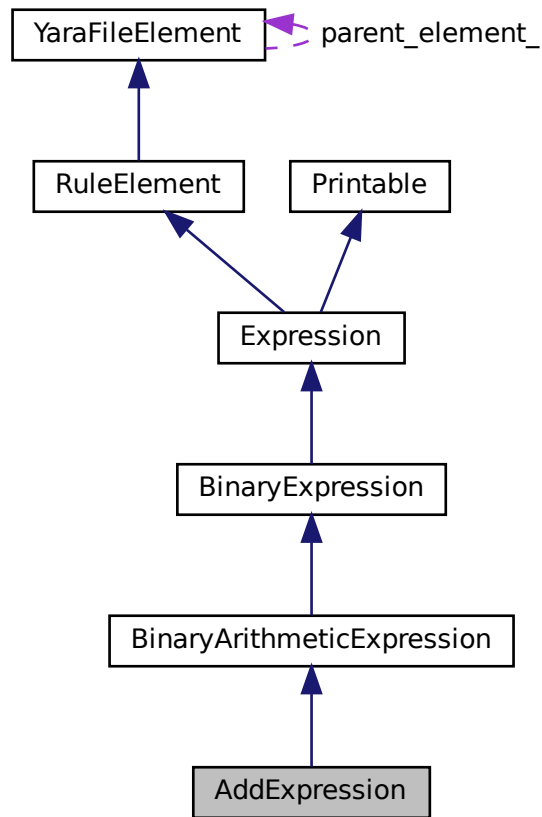
Class Documentation

5.1 AddExpression Class Reference

Inheritance diagram for AddExpression:



Collaboration diagram for AddExpression:



Public Member Functions

- **AddExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.1.1 Member Function Documentation

5.1.1.1 `accept()`

```
void AddExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.1.1.2 opsign()

```
std::string AddExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

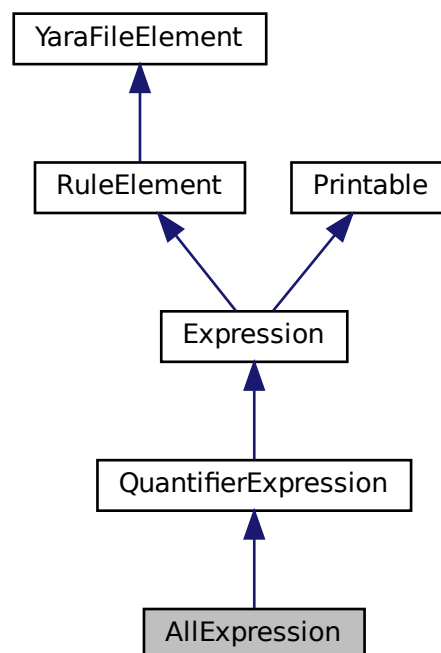
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.2 AllExpression Class Reference

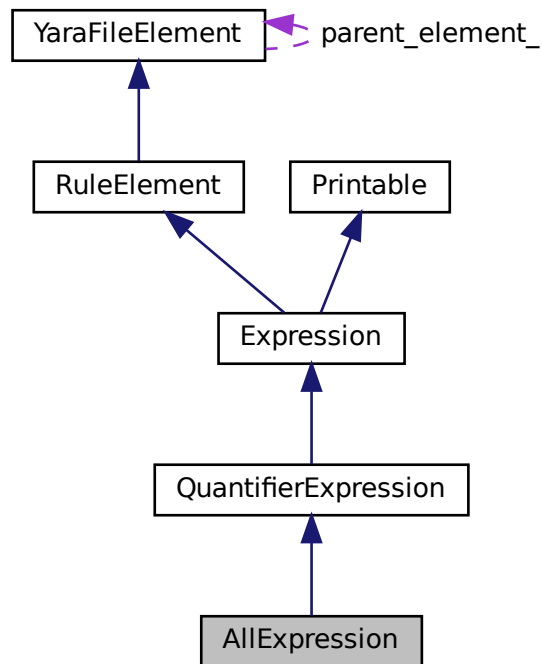
Represents 'all' keyword.

```
#include <expression.h>
```

Inheritance diagram for AllExpression:



Collaboration diagram for AllExpression:



Public Member Functions

- `std::stringstream` [getTextFormatted](#) () const override
- `void` [accept](#) ([Visitor](#) *v) override

Additional Inherited Members

5.2.1 Detailed Description

Represents 'all' keyword.

5.2.2 Member Function Documentation

5.2.2.1 accept()

```
void AllExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.2.2.2 getTextFormatted()

```
std::stringstream AllExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

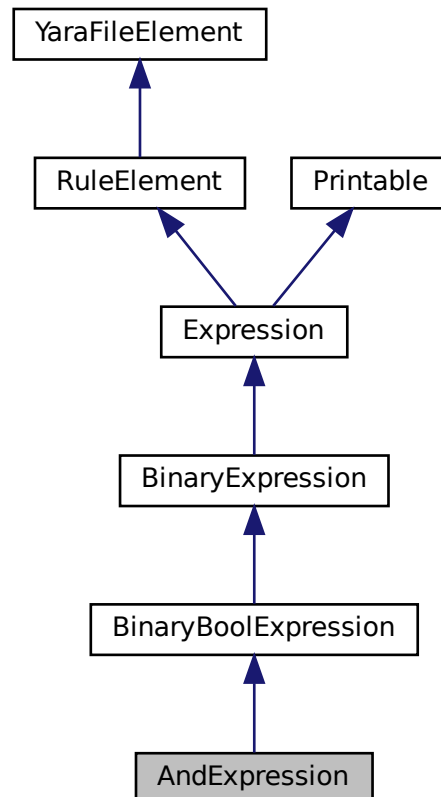
Implements [Printable](#).

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

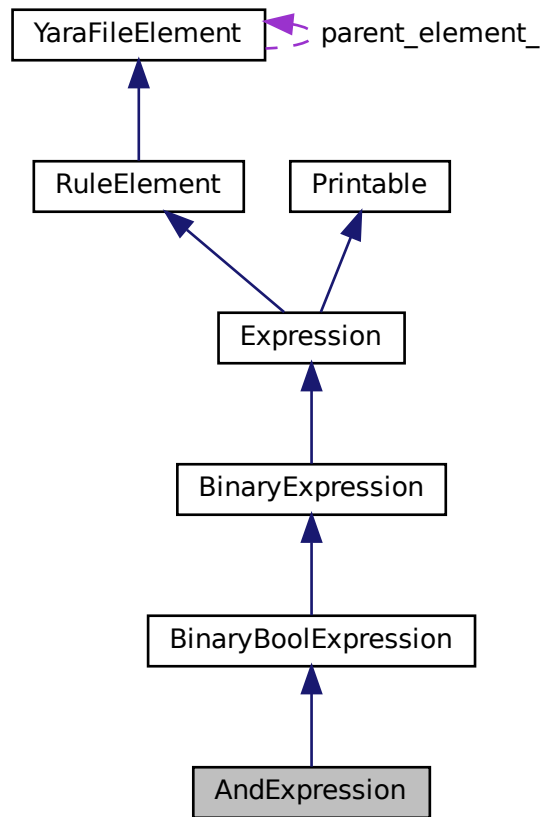
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.3 AndExpression Class Reference

Inheritance diagram for AndExpression:



Collaboration diagram for AndExpression:



Public Member Functions

- **AndExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.3.1 Member Function Documentation

5.3.1.1 `accept()`

```
void AndExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.3.1.2 opsign()

```
std::string AndExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

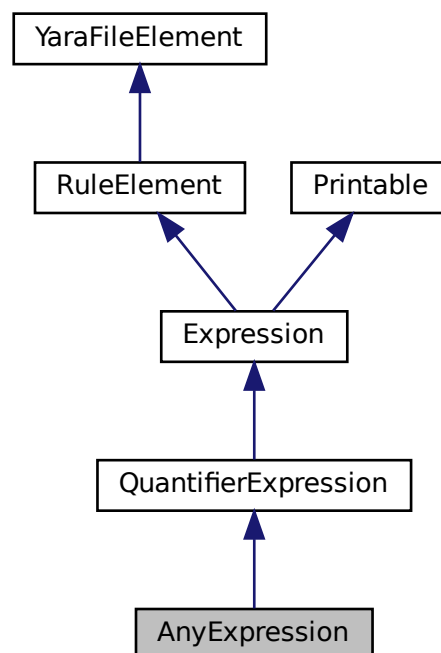
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.4 AnyExpression Class Reference

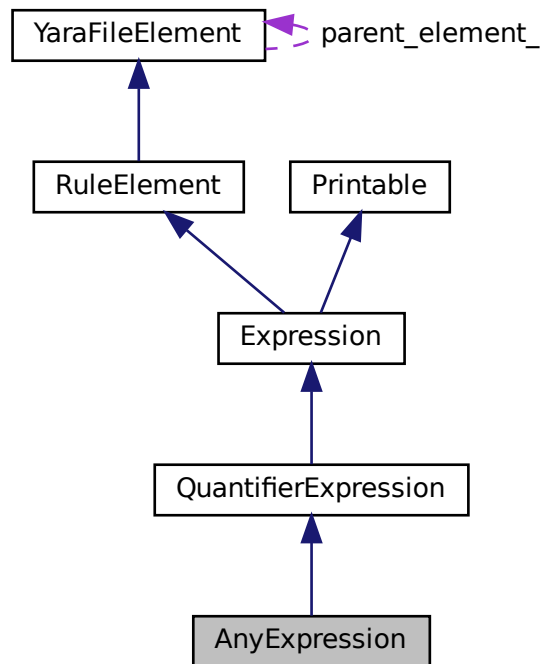
Represents 'any' keyword.

```
#include <expression.h>
```

Inheritance diagram for AnyExpression:



Collaboration diagram for AnyExpression:



Public Member Functions

- `std::stringstream` [getTextFormatted](#) () const override
- `void` [accept](#) ([Visitor](#) *v) override

Additional Inherited Members

5.4.1 Detailed Description

Represents 'any' keyword.

5.4.2 Member Function Documentation

5.4.2.1 accept()

```
void AnyExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.4.2.2 getTextFormatted()

```
std::stringstream AnyExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

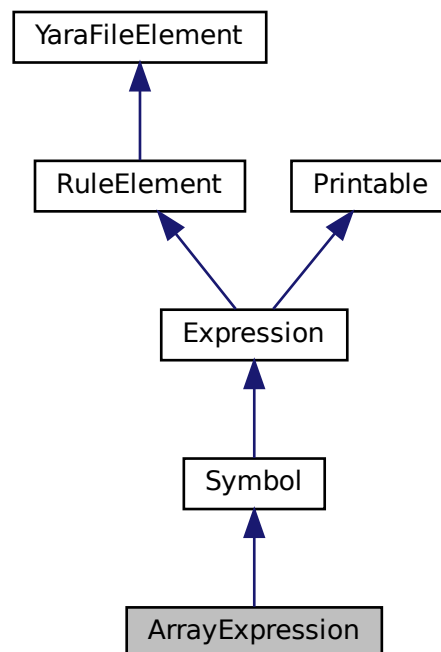
Implements [Printable](#).

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

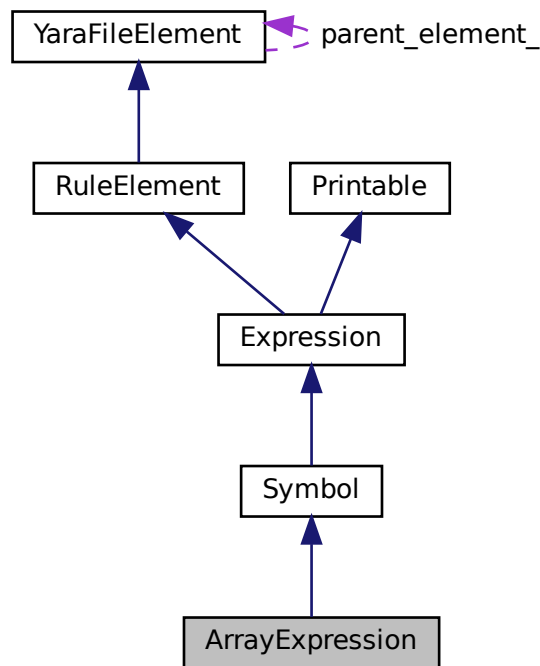
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.5 ArrayExpression Class Reference

Inheritance diagram for ArrayExpression:



Collaboration diagram for ArrayExpression:



Public Member Functions

- **ArrayExpression** (std::shared_ptr< [Symbol](#) > array, ExpressionPtr key)
- bool [isComplete](#) () const override
- bool [areOperandsValid](#) () const override
- bool [isValid](#) (std::string_view &msg) const override
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) (Visitor *v) override
- const std::shared_ptr< [Symbol](#) > & **getArray** () const
- const ExpressionPtr & **getKey** () const

Additional Inherited Members

5.5.1 Member Function Documentation

5.5.1.1 accept()

```
void ArrayExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.5.1.2 areOperandsValid()

```
bool ArrayExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Reimplemented from [Symbol](#).

5.5.1.3 getTextFormatted()

```
std::stringstream ArrayExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Reimplemented from [Symbol](#).

5.5.1.4 isComplete()

```
bool ArrayExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Reimplemented from [Symbol](#).

5.5.1.5 isValid()

```
bool ArrayExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

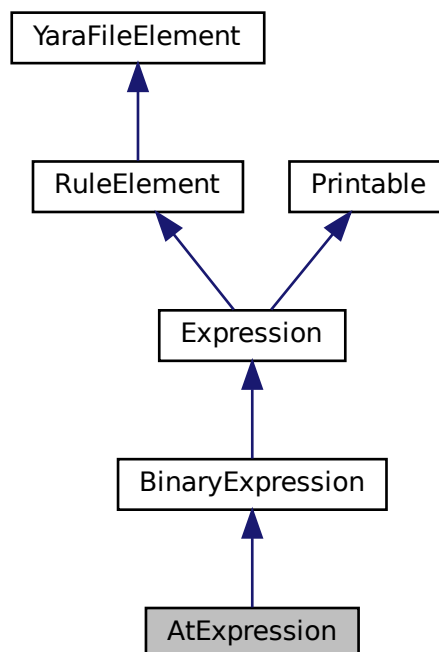
Reimplemented from [Symbol](#).

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

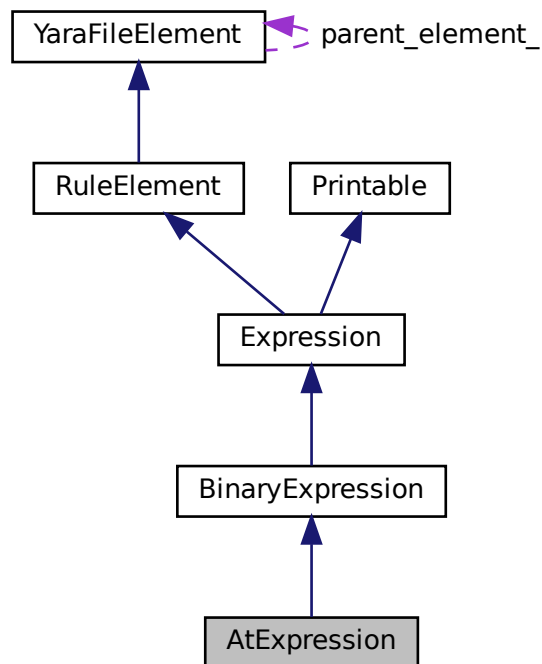
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.6 AtExpression Class Reference

Inheritance diagram for AtExpression:



Collaboration diagram for AtExpression:



Public Member Functions

- **AtExpression** (ExpressionPtr lop, ExpressionPtr rop)
- bool **isValid** (std::string_view &msg) const override
- **Expression::Type** **getType** () const override
- std::string **opsign** () const override
- void **accept** (Visitor *v) override

Additional Inherited Members

5.6.1 Member Function Documentation

5.6.1.1 accept()

```
void AtExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.6.1.2 getType()

```
Expression::Type AtExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.6.1.3 isValid()

```
bool AtExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

5.6.1.4 opsign()

```
std::string AtExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

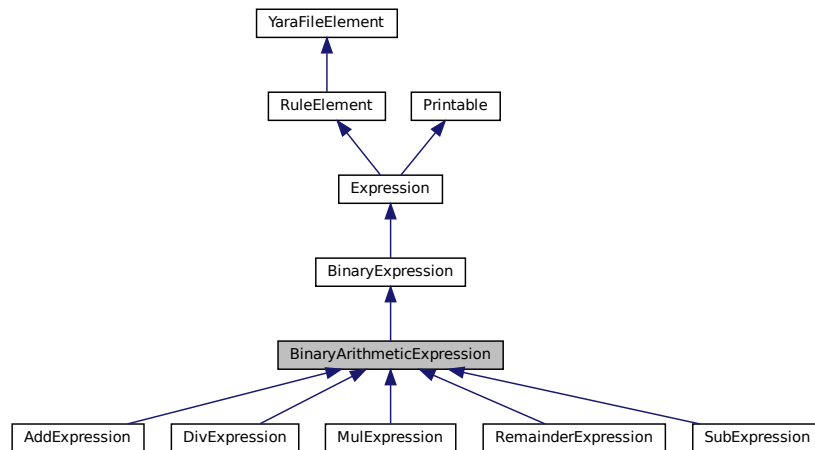
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.7 BinaryArithmeticExpression Class Reference

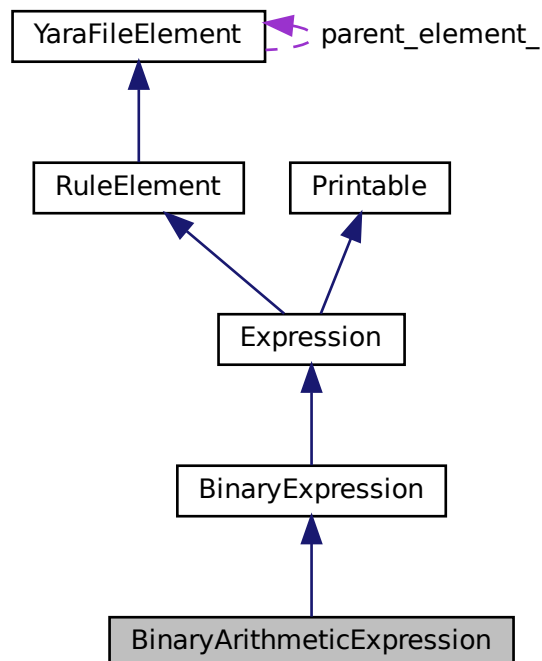
Groups all binary arithmetics operators for more generic type checking.

```
#include <expression.h>
```

Inheritance diagram for BinaryArithmeticExpression:



Collaboration diagram for BinaryArithmeticExpression:



Public Member Functions

- **BinaryArithmeticExpression** (ExpressionPtr lop, ExpressionPtr rop)
- virtual bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override

Additional Inherited Members

5.7.1 Detailed Description

Groups all binary arithmetics operators for more generic type checking.

5.7.2 Member Function Documentation

5.7.2.1 [getType\(\)](#)

```
Expression::Type BinaryArithmeticExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

Reimplemented in [RemainderExpression](#).

5.7.2.2 [isValid\(\)](#)

```
bool BinaryArithmeticExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

Reimplemented in [RemainderExpression](#).

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

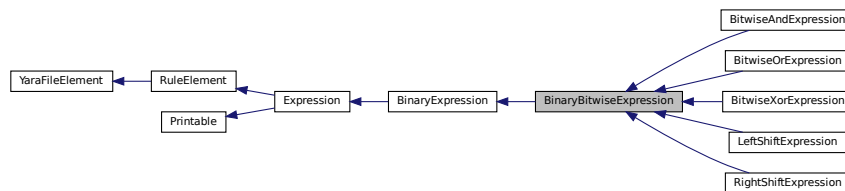
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.8 BinaryBitwiseExpression Class Reference

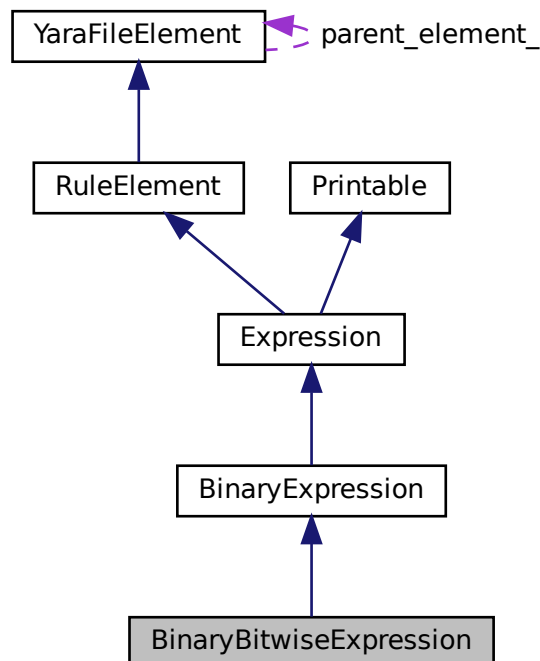
Groups all binary bitwise operators for more generic type checking.

```
#include <expression.h>
```

Inheritance diagram for BinaryBitwiseExpression:



Collaboration diagram for BinaryBitwiseExpression:



Public Member Functions

- **BinaryBitwiseExpression** (ExpressionPtr lop, ExpressionPtr rop)
- virtual bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override

Additional Inherited Members

5.8.1 Detailed Description

Groups all binary bitwise operators for more generic type checking.

5.8.2 Member Function Documentation

5.8.2.1 [getType\(\)](#)

```
Expression::Type BinaryBitwiseExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.8.2.2 [isValid\(\)](#)

```
bool BinaryBitwiseExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

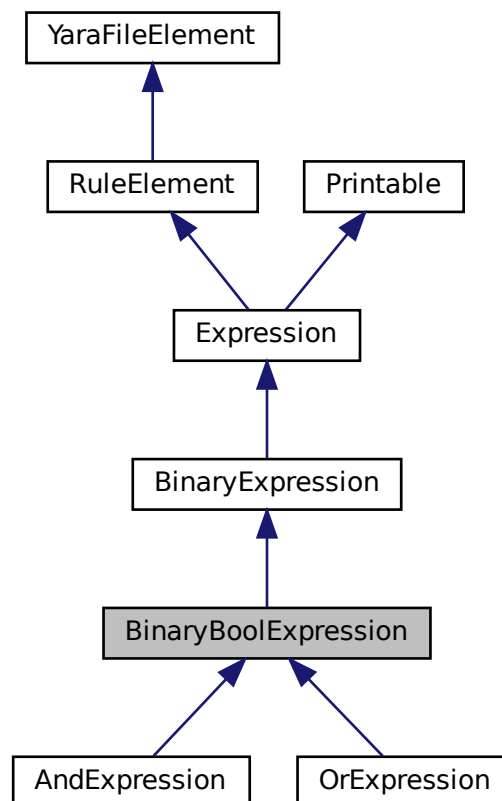
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.9 BinaryBoolExpression Class Reference

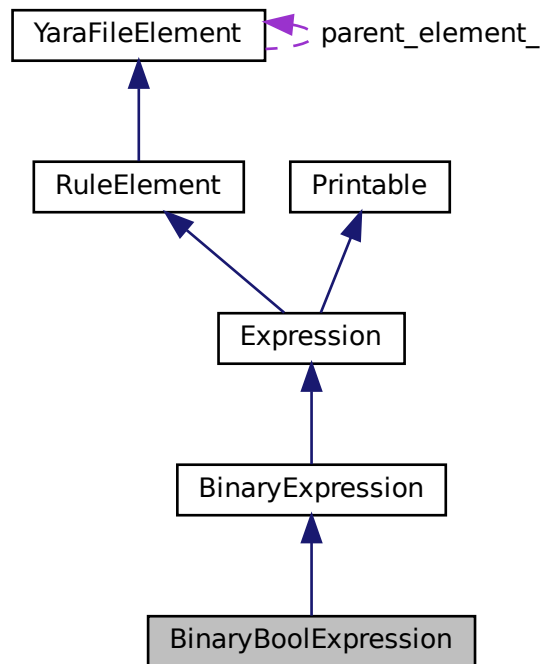
Groups all binary bitwise operators for more generic type checking.

```
#include <expression.h>
```

Inheritance diagram for BinaryBoolExpression:



Collaboration diagram for BinaryBoolExpression:



Public Member Functions

- **BinaryBoolExpression** (ExpressionPtr lop, ExpressionPtr rop)
- virtual bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override

Additional Inherited Members

5.9.1 Detailed Description

Groups all binary bitwise operators for more generic type checking.

5.9.2 Member Function Documentation

5.9.2.1 [getType\(\)](#)

```
Expression::Type BinaryBoolExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.9.2.2 isValid()

```
bool BinaryBoolExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

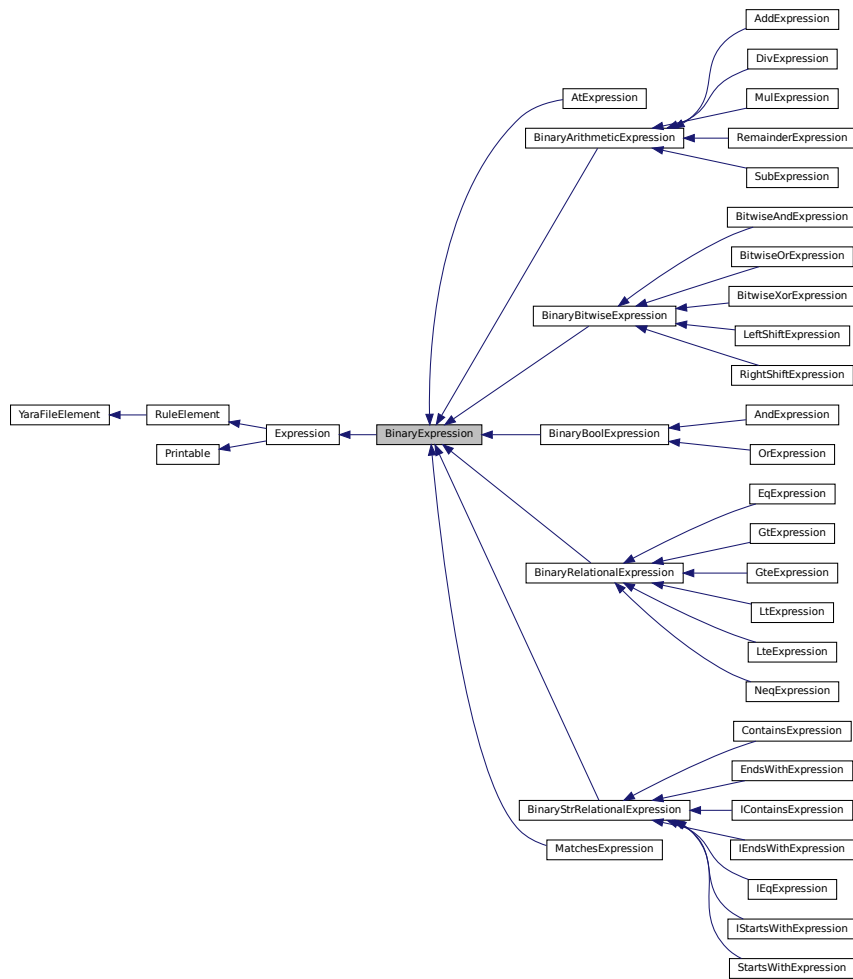
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.10 BinaryExpression Class Reference

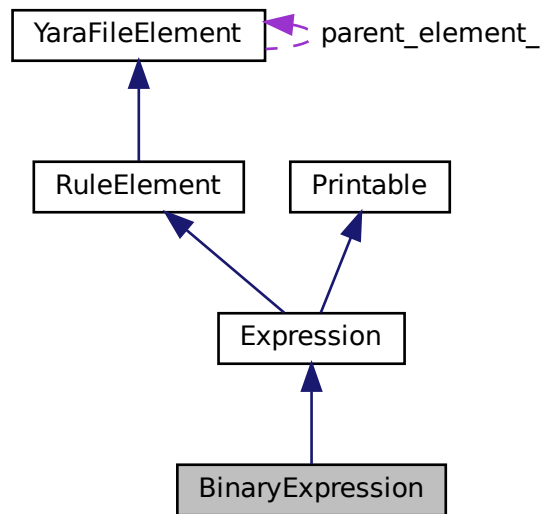
Class, that represents binary operators.

```
#include <expression.h>
```

Inheritance diagram for BinaryExpression:



Collaboration diagram for BinaryExpression:



Public Member Functions

- **BinaryExpression** (ExpressionPtr lop, ExpressionPtr rop)
- bool [isComplete](#) () const override
- bool [areOperandsValid](#) () const override
- virtual std::string [opsign](#) () const =0
- std::stringstream [getTextFormatted](#) () const override
- const ExpressionPtr & **getLop** () const
- const ExpressionPtr & **getRop** () const

Protected Attributes

- ExpressionPtr **lop_** = nullptr
- ExpressionPtr **rop_** = nullptr

Additional Inherited Members

5.10.1 Detailed Description

Class, that represents binary operators.

Base class for binary operators such as +,-,\,* etc.

5.10.2 Member Function Documentation

5.10.2.1 areOperandsValid()

```
bool BinaryExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.10.2.2 getTextFormatted()

```
std::stringstream BinaryExpression::getTextFormatted ( ) const [override], [virtual]
```

Returns stringstream with printable representation of expression. If there is invalid operand, it is missing in output stringstream.

Implements [Printable](#).

5.10.2.3 isComplete()

```
bool BinaryExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.10.2.4 opsign()

```
virtual std::string BinaryExpression::opsign ( ) const [inline], [pure virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implemented in [OrExpression](#), [AndExpression](#), [IEqExpression](#), [IEndsWithExpression](#), [EndsWithExpression](#), [IStartsWithExpression](#), [StartsWithExpression](#), [IContainsExpression](#), [ContainsExpression](#), [NeqExpression](#), [EqExpression](#), [GteExpression](#), [GtExpression](#), [LteExpression](#), [LtExpression](#), [BitwiseXorExpression](#), [BitwiseAndExpression](#), [BitwiseOrExpression](#), [RightShiftExpression](#), [LeftShiftExpression](#), [DivExpression](#), [RemainderExpression](#), [MulExpression](#), [SubExpression](#), [AddExpression](#), [AtExpression](#), and [MatchesExpression](#).

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

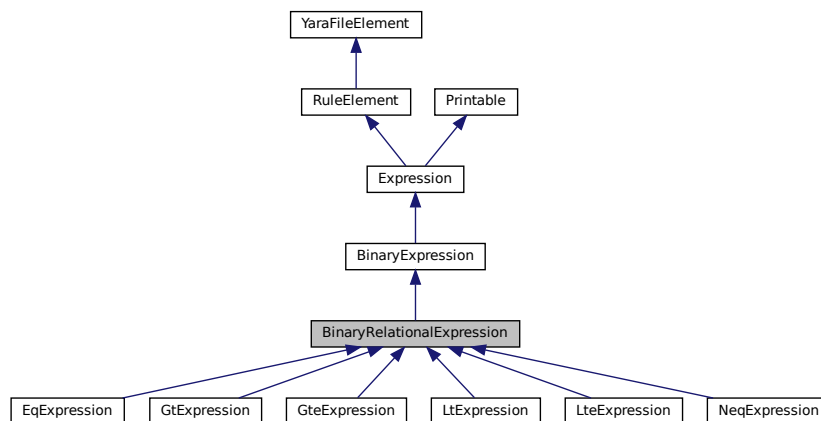
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.11 BinaryRelationalExpression Class Reference

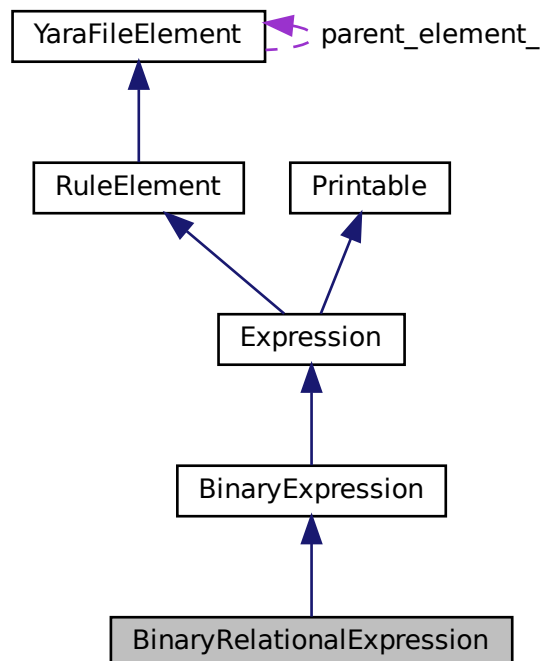
Groups all binary bitwise operators for more generic type checking.

```
#include <expression.h>
```

Inheritance diagram for BinaryRelationalExpression:



Collaboration diagram for BinaryRelationalExpression:



Public Member Functions

- **BinaryRelationalExpression** (ExpressionPtr lop, ExpressionPtr rop)
- virtual bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override

Additional Inherited Members

5.11.1 Detailed Description

Groups all binary bitwise operators for more generic type checking.

5.11.2 Member Function Documentation

5.11.2.1 [getType](#)()

```
Expression::Type BinaryRelationalExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.11.2.2 [isValid](#)()

```
bool BinaryRelationalExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

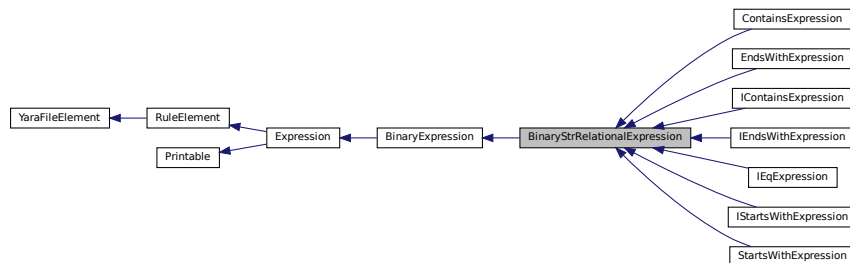
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.12 BinaryStrRelationalExpression Class Reference

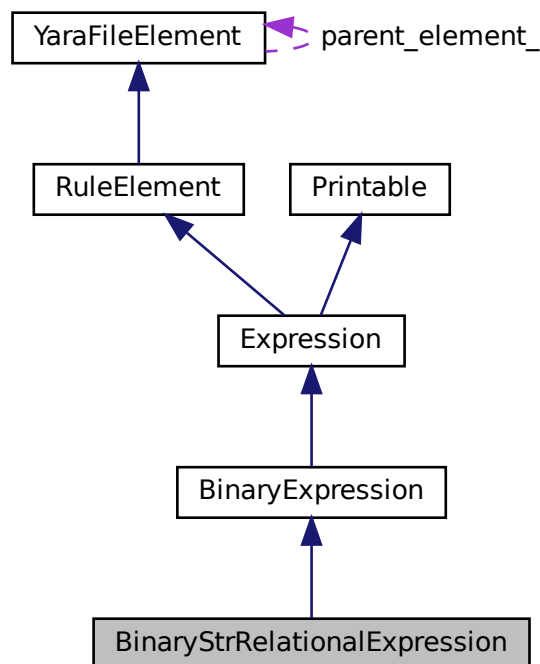
Groups all binary bitwise operators for more generic type checking.

```
#include <expression.h>
```

Inheritance diagram for BinaryStrRelationalExpression:



Collaboration diagram for BinaryStrRelationalExpression:



Public Member Functions

- **BinaryStrRelationalExpression** (ExpressionPtr lop, ExpressionPtr rop)
- virtual bool **isValid** (std::string_view &msg) const override
- [Expression::Type](#) **getType** () const override

Additional Inherited Members

5.12.1 Detailed Description

Groups all binary bitwise operators for more generic type checking.

5.12.2 Member Function Documentation

5.12.2.1 `getType()`

```
Expression::Type BinaryStrRelationalExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.12.2.2 `isValid()`

```
bool BinaryStrRelationalExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

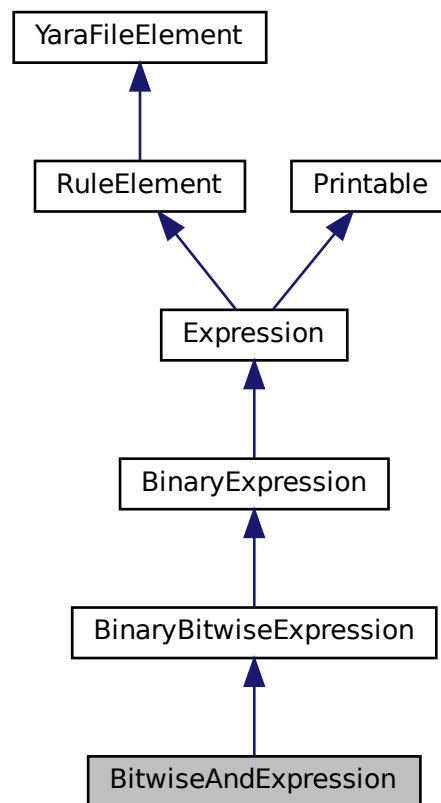
Implements [Expression](#).

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

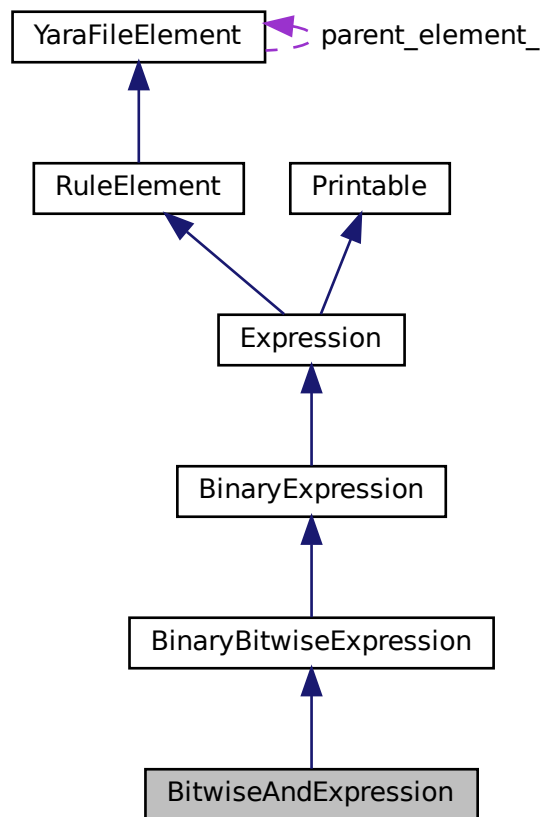
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.13 BitwiseAndExpression Class Reference

Inheritance diagram for BitwiseAndExpression:



Collaboration diagram for BitwiseAndExpression:



Public Member Functions

- **BitwiseAndExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.13.1 Member Function Documentation

5.13.1.1 [accept\(\)](#)

```
void BitwiseAndExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.13.1.2 opsign()

```
std::string BitwiseAndExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

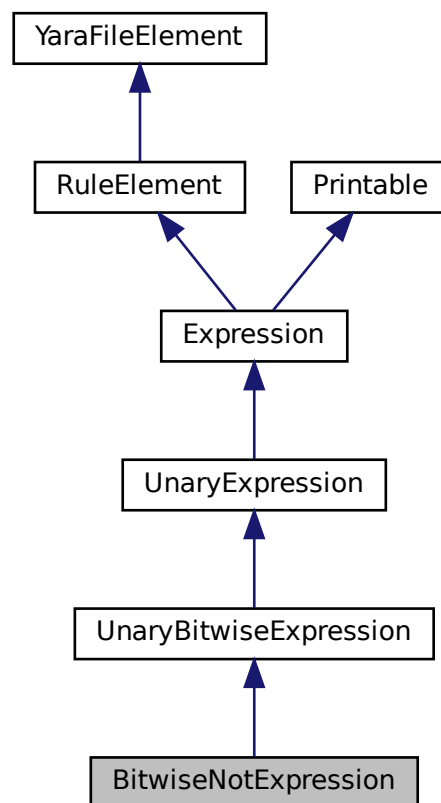
Implements [BinaryExpression](#).

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

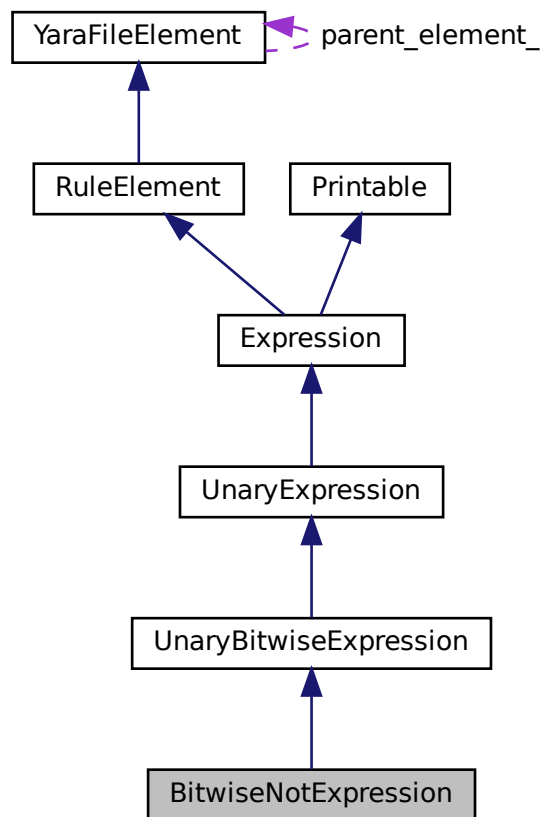
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.14 BitwiseNotExpression Class Reference

Inheritance diagram for BitwiseNotExpression:



Collaboration diagram for BitwiseNotExpression:



Public Member Functions

- **BitwiseNotExpression** (ExpressionPtr op)
- std::string **opsign** () const override
- void **accept** (Visitor *v) override

Additional Inherited Members

5.14.1 Member Function Documentation

5.14.1.1 accept()

```
void BitwiseNotExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

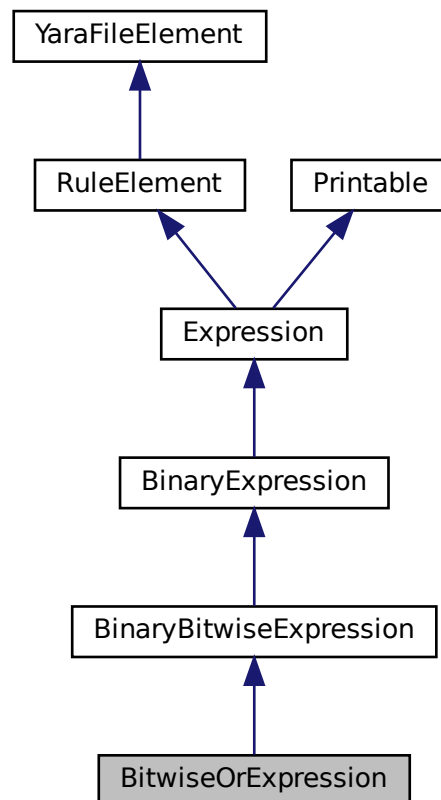
Implements [Expression](#).

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

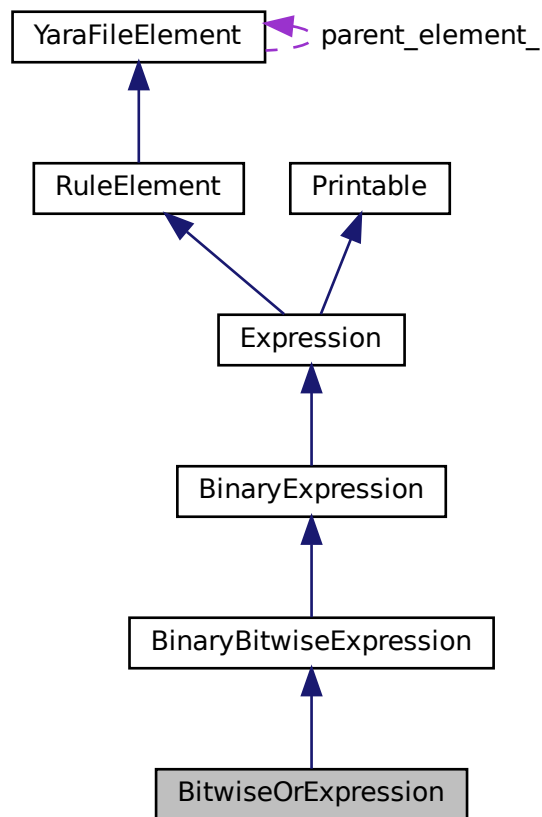
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.15 BitwiseOrExpression Class Reference

Inheritance diagram for BitwiseOrExpression:



Collaboration diagram for BitwiseOrExpression:



Public Member Functions

- **BitwiseOrExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) ([Visitor](#) *v) override

Additional Inherited Members

5.15.1 Member Function Documentation

5.15.1.1 [accept\(\)](#)

```
void BitwiseOrExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.15.1.2 opsign()

```
std::string BitwiseOrExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

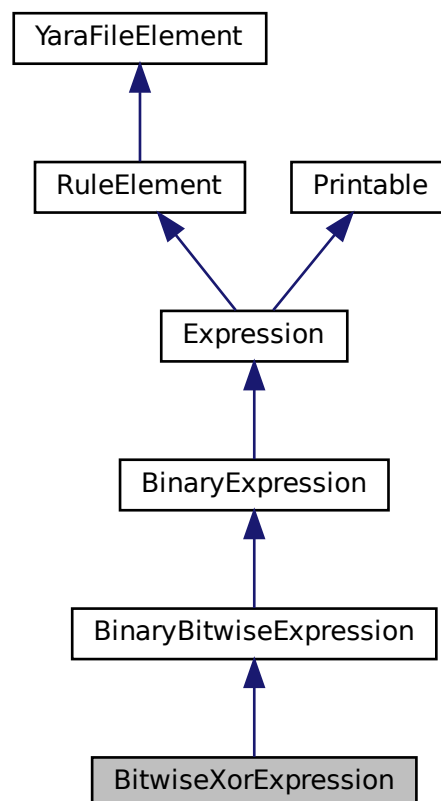
Implements [BinaryExpression](#).

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

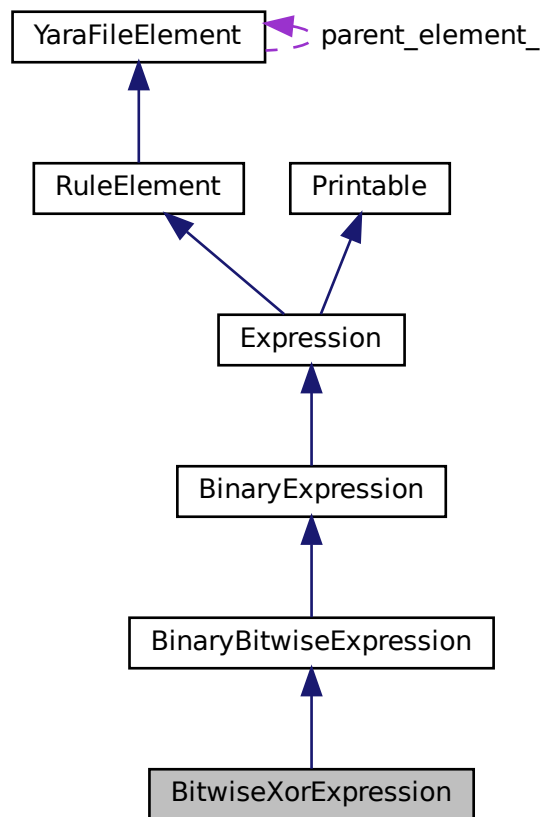
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.16 BitwiseXorExpression Class Reference

Inheritance diagram for BitwiseXorExpression:



Collaboration diagram for BitwiseXorExpression:



Public Member Functions

- **BitwiseXorExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.16.1 Member Function Documentation

5.16.1.1 [accept\(\)](#)

```
void BitwiseXorExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.16.1.2 opsign()

```
std::string BitwiseXorExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

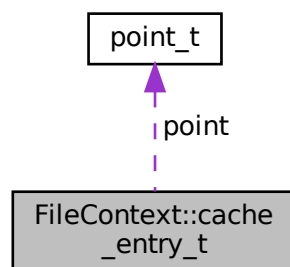
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.17 FileContext::cache_entry_t Struct Reference

Entry in stored in cache.

```
#include <file_context.h>
```

Collaboration diagram for FileContext::cache_entry_t:



Public Attributes

- `offset_t offset`
- `point_t point`

5.17.1 Detailed Description

Entry in stored in cache.

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

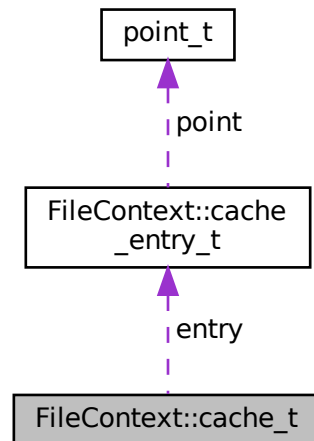
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/file_context.h](#)

5.18 FileContext::cache_t Struct Reference

Cache of [FileContext](#) for offset_t <-> [point_t](#) conversion.

```
#include <file_context.h>
```

Collaboration diagram for FileContext::cache_t:



Public Attributes

- [FileContext::cache_entry_t](#) entry
- bool **is_valid** = false

5.18.1 Detailed Description

Cache of [FileContext](#) for offset_t <-> [point_t](#) conversion.

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

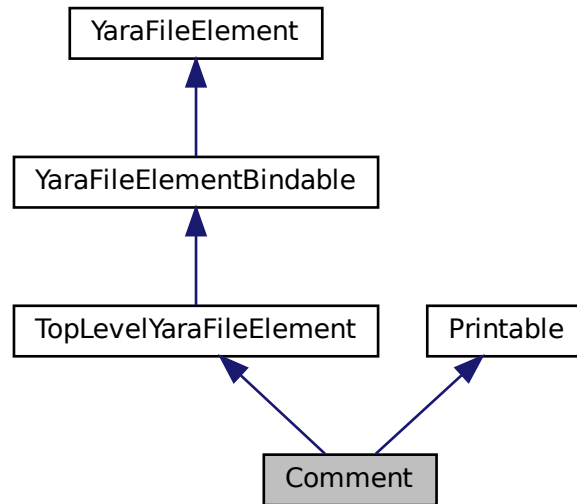
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[file_context.h](#)

5.19 Comment Class Reference

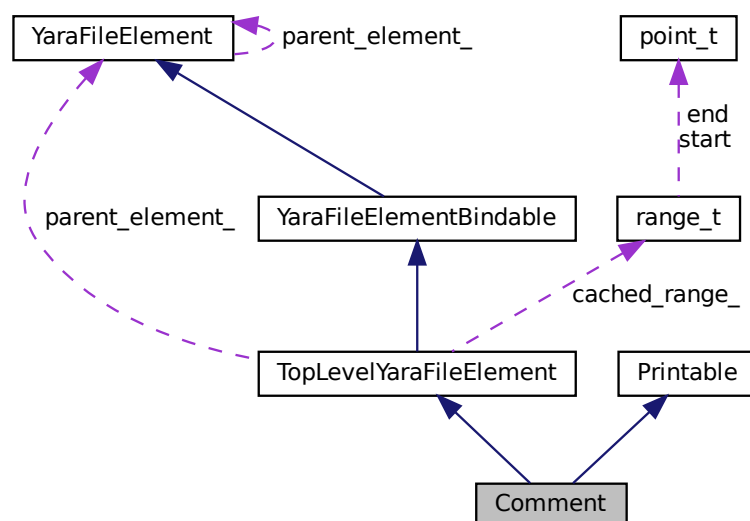
Class representing comments (all valid types in YARA language)

```
#include <other.h>
```

Inheritance diagram for Comment:



Collaboration diagram for Comment:



Public Member Functions

- **Comment** (const std::string &text, const offset_t &pos)
- const std::string & **getText** () const
- std::stringstream **getTextFormatted** () const override

Additional Inherited Members

5.19.1 Detailed Description

Class representing comments (all valid types in YARA language)

5.19.2 Member Function Documentation

5.19.2.1 getTextFormatted()

```
std::stringstream Comment::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

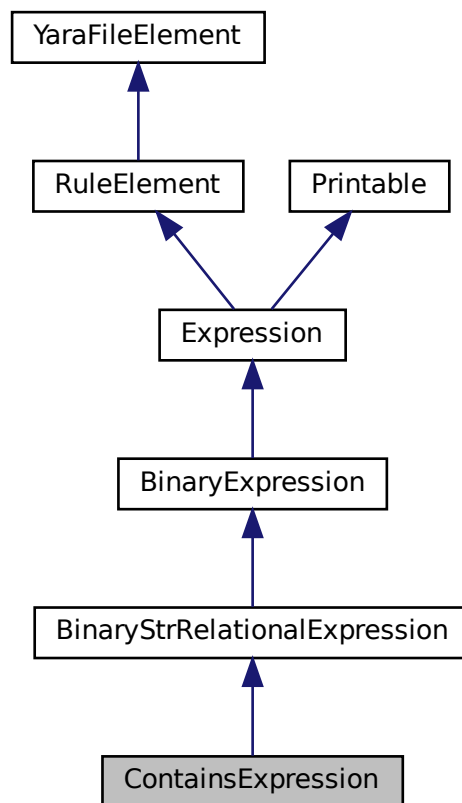
Implements [Printable](#).

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

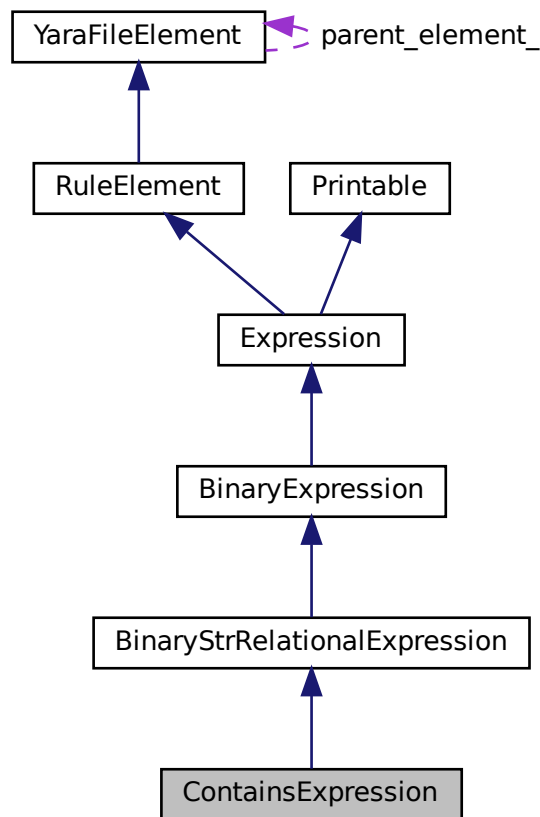
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[other.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[other.cpp](#)

5.20 ContainsExpression Class Reference

Inheritance diagram for ContainsExpression:



Collaboration diagram for ContainsExpression:



Public Member Functions

- **ContainsExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.20.1 Member Function Documentation

5.20.1.1 [accept\(\)](#)

```
void ContainsExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.20.1.2 opsign()

```
std::string ContainsExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

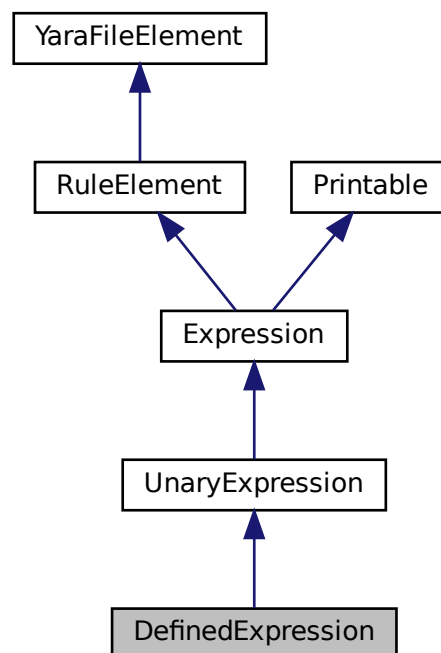
Implements [BinaryExpression](#).

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

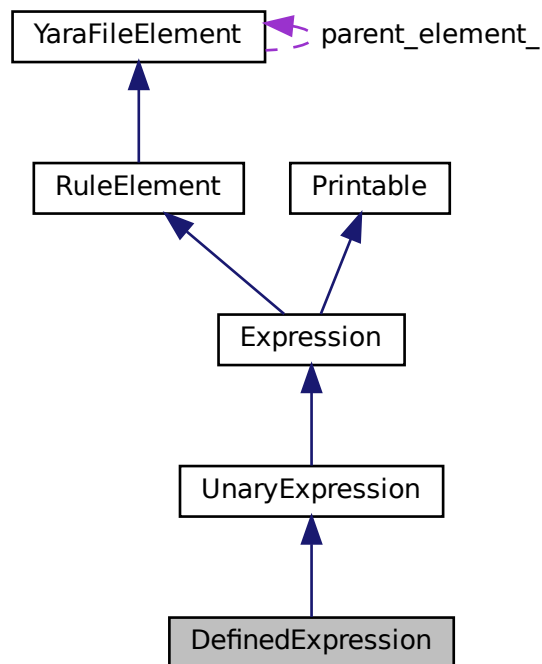
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.21 DefinedExpression Class Reference

Inheritance diagram for DefinedExpression:



Collaboration diagram for DefinedExpression:



Public Member Functions

- **DefinedExpression** (ExpressionPtr op)
- bool **isValid** (std::string_view &msg) const override
- **Expression::Type** **getType** () const override
- std::string **opsign** () const override
- void **accept** (Visitor *v) override

Additional Inherited Members

5.21.1 Member Function Documentation

5.21.1.1 accept()

```
void DefinedExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.21.1.2 getType()

```
Expression::Type DefinedExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.21.1.3 isValid()

```
bool DefinedExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

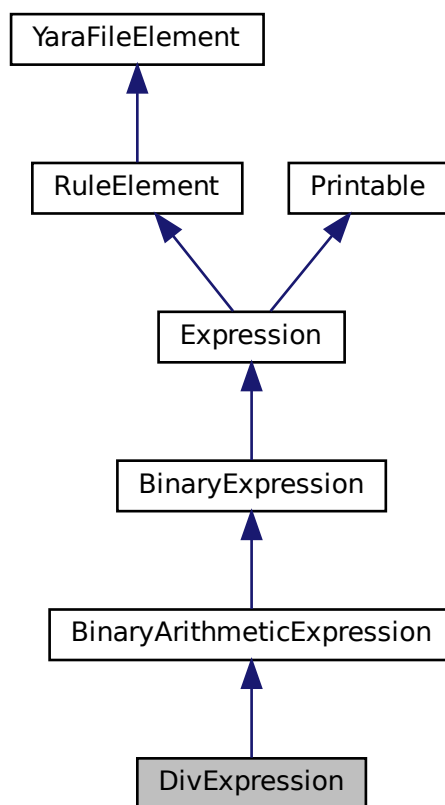
Implements [Expression](#).

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

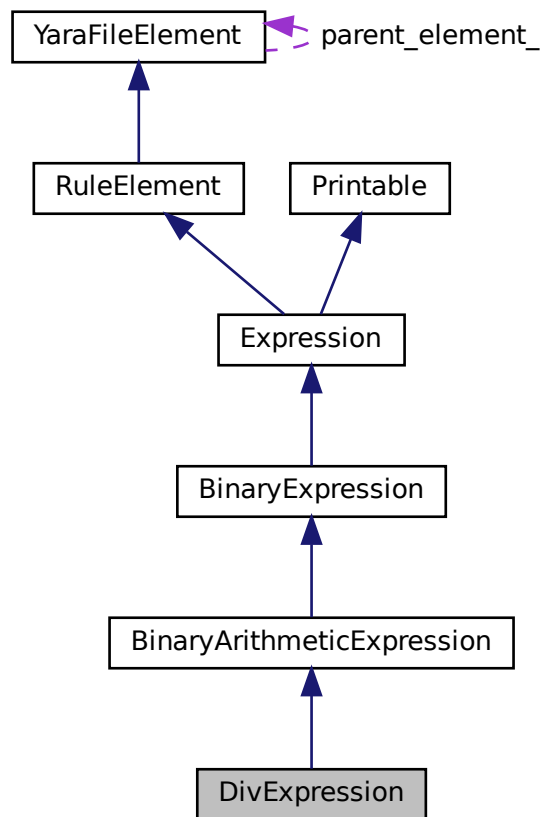
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.22 DivExpression Class Reference

Inheritance diagram for DivExpression:



Collaboration diagram for DivExpression:



Public Member Functions

- **DivExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.22.1 Member Function Documentation

5.22.1.1 `accept()`

```
void DivExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.22.1.2 opsign()

```
std::string DivExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.23 edit_t Struct Reference

Represents textual edit of [YaraFile](#).

```
#include <yara_file.h>
```

Public Member Functions

- **edit_t** (const offset_t &edit_offset, const size_t &edit_ins_n, const size_t &edit_del_n)
- **edit_t** (const offset_t &edit_offset, const size_t &edit_ins_n, const size_t &edit_del_n, const std::string &text)

Public Attributes

- offset_t [offset](#)
Offset where the edit begins (the first affected character)
- size_t [ins_n](#)
Number of inserted characters.
- size_t [del_n](#)
Number of deleted characters.
- std::string [new_text](#) = {}
The inserted text (optional, recommended)
- TSInputEdit **ts_edit**
- bool [fast_edit](#) = false
Flag if edit is specified only by numeric values (new text is not needed) - not recommended.

5.23.1 Detailed Description

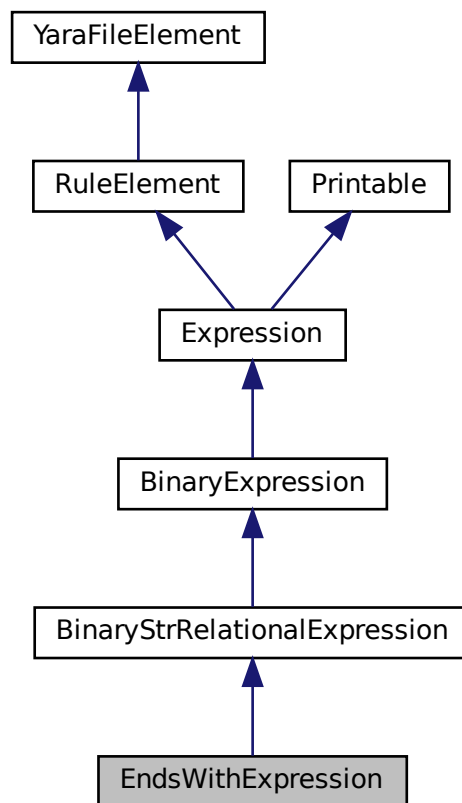
Represents textual edit of [YaraFile](#).

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

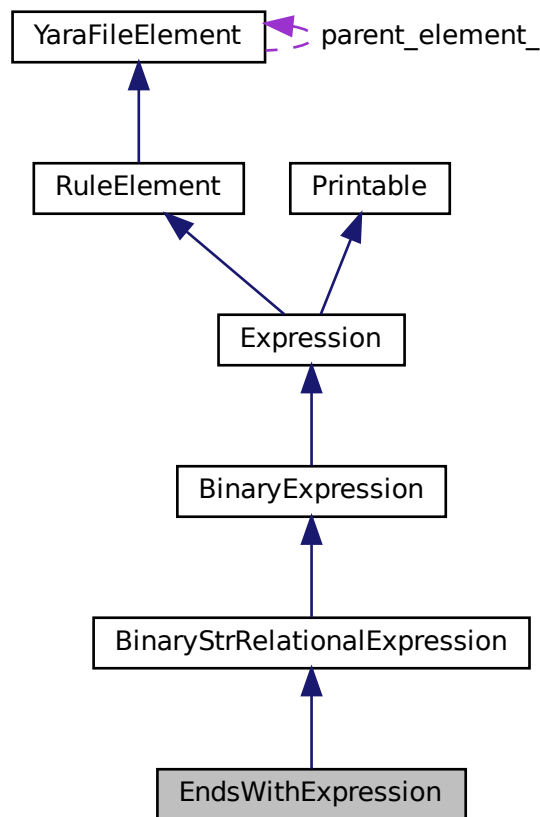
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[yara_file.h](#)

5.24 EndsWithExpression Class Reference

Inheritance diagram for EndsWithExpression:



Collaboration diagram for EndsWithExpression:



Public Member Functions

- **EndsWithExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.24.1 Member Function Documentation

5.24.1.1 `accept()`

```
void EndsWithExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.24.1.2 opsign()

```
std::string EndsWithExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

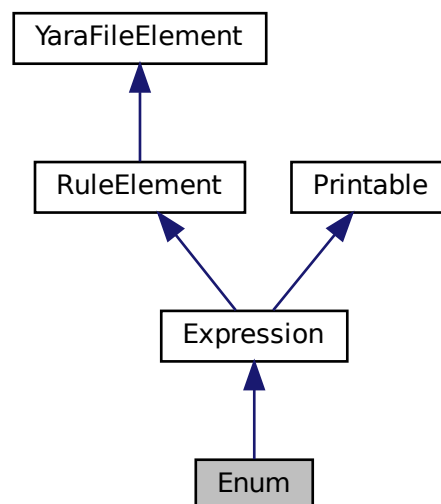
Implements [BinaryExpression](#).

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

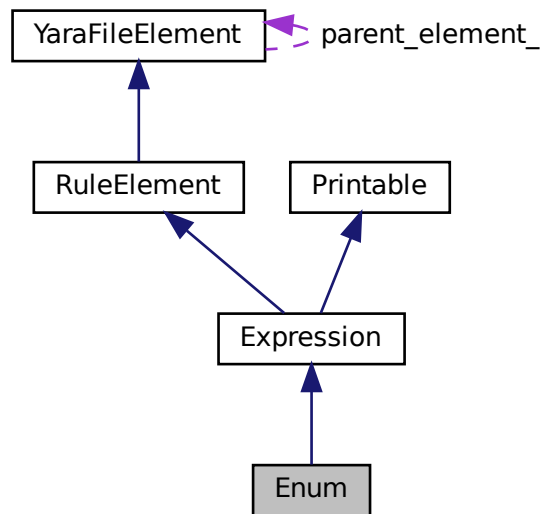
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.25 Enum Class Reference

Inheritance diagram for Enum:



Collaboration diagram for Enum:



Public Member Functions

- **Enum** (`std::vector< ExpressionPtr > values`)
- `bool areOperandsValid ()` const override
- `bool isComplete ()` const override
- `bool isValid (std::string_view &msg)` const override
- `Expression::Type getType ()` const override
- `std::stringstream getTextFormatted ()` const override
- `void accept (Visitor *v)` override
- `const std::vector< ExpressionPtr > &getValues ()` const

Additional Inherited Members

5.25.1 Member Function Documentation

5.25.1.1 accept()

```
void Enum::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.25.1.2 areOperandsValid()

```
bool Enum::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.25.1.3 getTextFormatted()

```
std::stringstream Enum::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.25.1.4 getType()

```
Expression::Type Enum::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.25.1.5 isComplete()

```
bool Enum::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.25.1.6 isValid()

```
bool Enum::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

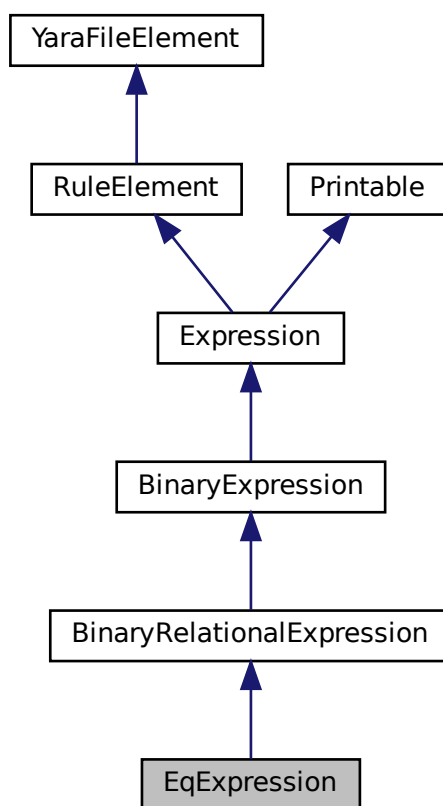
Implements [Expression](#).

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

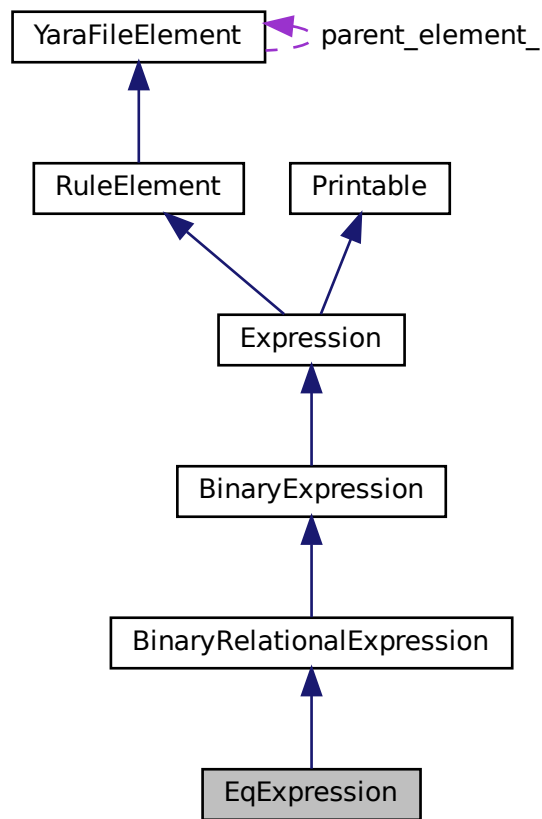
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.26 EqExpression Class Reference

Inheritance diagram for EqExpression:



Collaboration diagram for EqExpression:



Public Member Functions

- **EqExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.26.1 Member Function Documentation

5.26.1.1 [accept\(\)](#)

```
void EqExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.26.1.2 opsign()

```
std::string EqExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

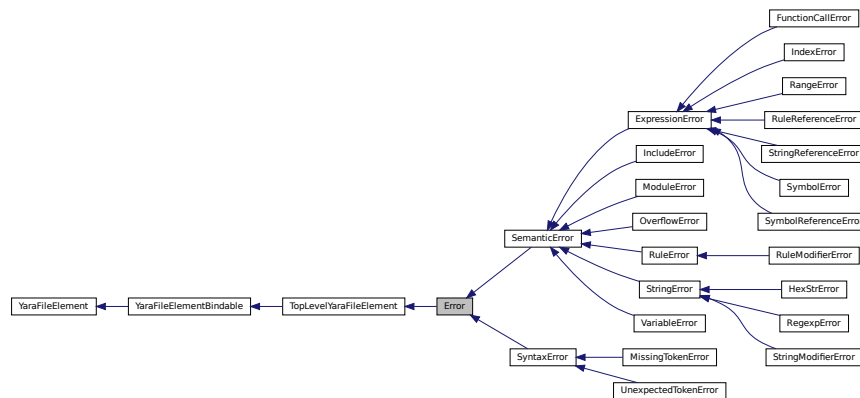
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.27 Error Class Reference

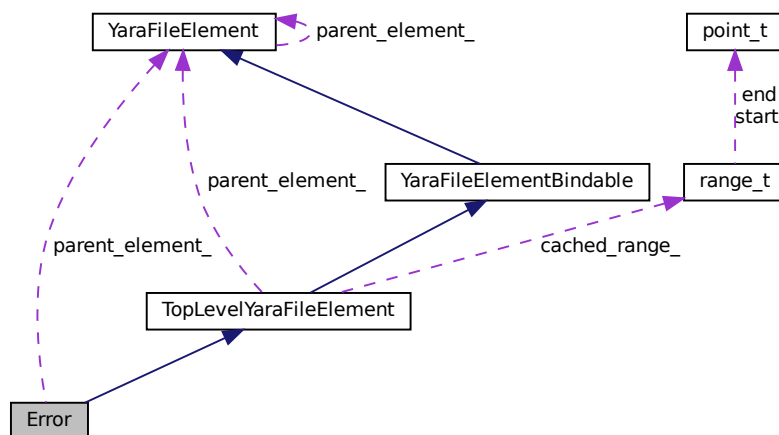
Class, that represents errors in analyzed source code. The main difference between `ErrorExceptions` and `Errors` is in usage - `Errors` are stored in some vector/list or in something and `ErrorException` are thrown.

```
#include <error.h>
```

Inheritance diagram for `Error`:



Collaboration diagram for `Error`:



Public Member Functions

- **Error** (const std::string &desc, const offset_t &pos, const size_t &len)
- const std::string & [getDescription](#) () const
- [ErrorException](#) [exception](#) () const

Protected Attributes

- std::string [desc_](#)
Textual information about error.
- [YaraFileElement](#) * [parent_element_](#) = nullptr
Parent element of an error.

5.27.1 Detailed Description

Class, that represents errors in analyzed source code. The main difference between Exceptions and Errors is in usage - Errors are stored in some vector/list or in something and [ErrorException](#) are thrown.

5.27.2 Member Function Documentation

5.27.2.1 [exception\(\)](#)

```
ErrorException Error::exception ( ) const
```

Provides conversion to throwable exception

5.27.2.2 [getDescription\(\)](#)

```
const std::string & Error::getDescription ( ) const
```

Returns textual description of error - meaning of textual description depends on concrete type of error

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[error.cpp](#)

5.28 ErrorCollector< E > Class Template Reference

Class responsible for collecting all types of errors. It is basically a map of vectors. Vectors in map are indexed by the offset where errors occurred in [YaraFile](#).

```
#include <error_collector.h>
```

Public Member Functions

- [SymTabOffset](#)< E > & [get](#) ()
- const [SymTabOffset](#)< E > & [getConst](#) () const
- void [clear](#) ()
- void [add](#) (std::unique_ptr< E > new_error, [YaramodConfig::ErrorMode](#) mode)
- template<typename Fn >
uint32_t [remove](#) (const Fn &f)
- template<typename Fn >
uint32_t [remove](#) (const Fn &f, offset_t start, offset_t end)
- template<typename Fn >
uint32_t [revRemoveUntil](#) (const Fn &f, offset_t rstart)
- template<typename Fn >
void [update](#) (const Fn &f)
- template<typename Fn >
void [update](#) (const Fn &f, offset_t start, offset_t end)
- void [dump](#) (const std::string &string) const

5.28.1 Detailed Description

```
template<class E>
class ErrorCollector< E >
```

Class responsible for collecting all types of errors. It is basically a map of vectors. Vectors in map are indexed by the offset where errors occurred in [YaraFile](#).

5.28.2 Member Function Documentation

5.28.2.1 add()

```
template<class E >
void ErrorCollector< E >::add (
    std::unique_ptr< E > new_error,
    YaramodConfig::ErrorMode mode )
```

Adds an error to the map

5.28.2.2 clear()

```
template<class E >
void ErrorCollector< E >::clear
```

Clears the map with errors

5.28.2.3 dump()

```
template<class E >
void ErrorCollector< E >::dump (
    const std::string & string ) const
```

Prints all errors to stderr (mainly for debugging purposes)

5.28.2.4 get()

```
template<class E >
SymTabOffset< E > & ErrorCollector< E >::get
```

Returns pointer with the all errors, that were collected

5.28.2.5 getConst()

```
template<class E >
const SymTabOffset< E > & ErrorCollector< E >::getConst
```

Returns pointer with the all errors, that were collected

5.28.2.6 remove()

```
template<class E >
template<typename Fn >
uint32_t ErrorCollector< E >::remove (
    const Fn & f )
```

Removes errors from the map. Errors, that should be deleted, are specified by given bool function

Returns

Number of removed errors

5.28.2.7 revRemoveUntil()

```
template<class V >
template<typename Fn >
uint32_t ErrorCollector< V >::revRemoveUntil (
    const Fn & f,
    offset_t rstart )
```

Removes errors backwards from starting at offset

5.28.2.8 update()

```
template<class E >
template<typename Fn >
void ErrorCollector< E >::update (
    const Fn & f )
```

Updates errors. The key (error position/offset), can be also modified.

Returns

Number of updated errors

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

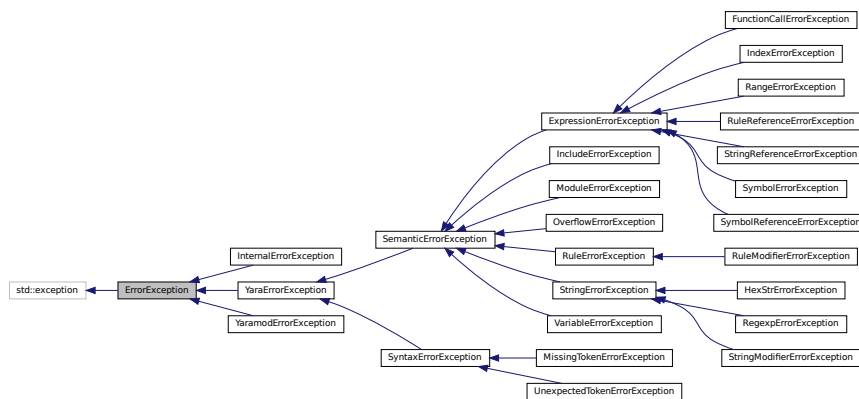
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error_collector.h

5.29 RecognitionException Class Reference

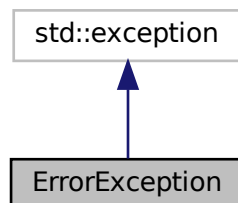
Base class for exceptions.

```
#include <error.h>
```

Inheritance diagram for RecognitionException:



Collaboration diagram for RecognitionException:



Public Member Functions

- **ErrorException** (const std::string &msg)
- virtual const char * **what** () const noexcept override

5.29.1 Detailed Description

Base class for exceptions.

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

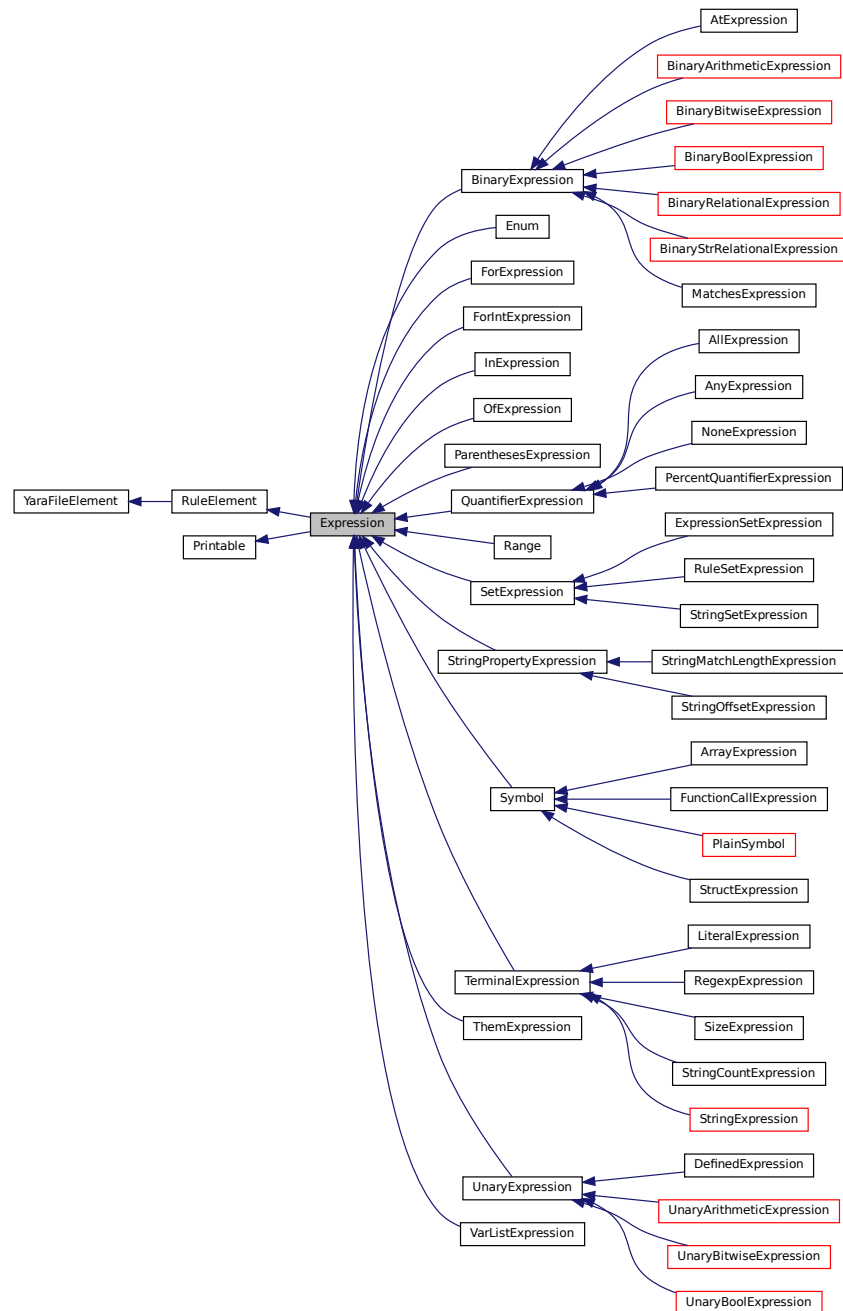
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[error.cpp](#)

5.30 Expression Class Reference

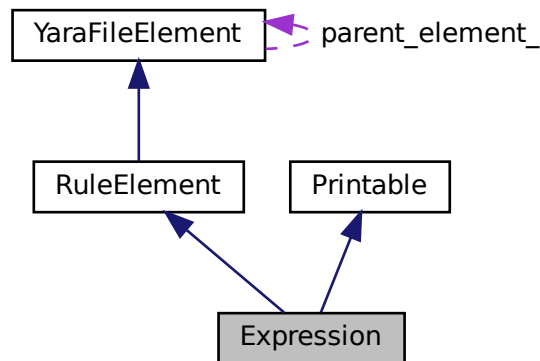
Represents expression in condition and in definition of internal variables.

```
#include <expression.h>
```

Inheritance diagram for Expression:



Collaboration diagram for Expression:



Public Types

- enum class [Type](#) {
Undefined = 0 , **Object** , **Bool** , **Int** ,
Float , **String** , **Regexp** }
Data type of expression (for semantics checks)

Public Member Functions

- virtual [Expression::Type](#) [getType](#) () const =0
- virtual bool [isValid](#) (std::string_view &msg) const =0
- virtual bool [areOperandsValid](#) () const =0
- virtual bool [isComplete](#) () const =0
- bool [isUndefined](#) () const
- bool [isObject](#) () const
- bool [isBool](#) () const
- bool [isInt](#) () const
- bool [isFloat](#) () const
- bool [isString](#) () const
- bool [isRegexp](#) () const
- virtual void [accept](#) ([Visitor](#) *v)=0

Static Public Member Functions

- static std::string_view [typeToString](#) ([Expression::Type](#) type)
- static bool [isArithmeticCompatible](#) (const ExpressionPtr &l, const ExpressionPtr &r)

Additional Inherited Members

5.30.1 Detailed Description

Represents expression in condition and in definition of internal variables.

It's base class for concrete expression classes, that represents corresponding type of expression. Because concrete classes have operands also of [Expression](#) type, [Expression](#) objects make up tree-like structure.

5.30.2 Member Function Documentation

5.30.2.1 `accept()`

```
virtual void Expression::accept (
    Visitor * v ) [pure virtual]
```

Accept method for visitor pattern

Implemented in [ForIntExpression](#), [VarListExpression](#), [ForExpression](#), [OfExpression](#), [ExpressionSetExpression](#), [RuleSetExpression](#), [StringSetExpression](#), [ThemExpression](#), [PercentQuantifierExpression](#), [AllExpression](#), [AnyExpression](#), [NoneExpression](#), [RuleWildcardExpression](#), [VariableExpression](#), [PlainSymbol](#), [FunctionCallExpression](#), [StructExpression](#), [ArrayExpression](#), [StringMatchLengthExpression](#), [StringOffsetExpression](#), [StringCountExpression](#), [StringWildcardExpression](#), [StringExpression](#), [RegexExpression](#), [SizeExpression](#), [LiteralExpression](#), [NotExpression](#), [BitwiseNotExpression](#), [UnaryMinusExpression](#), [DefinedExpression](#), [OrExpression](#), [AndExpression](#), [IEqExpression](#), [IEndsWithExpression](#), [EndsWithExpression](#), [IStartsWithExpression](#), [StartsWithExpression](#), [IContainsExpression](#), [ContainsExpression](#), [NeqExpression](#), [EqExpression](#), [GteExpression](#), [GtExpression](#), [LteExpression](#), [LtExpression](#), [BitwiseXorExpression](#), [BitwiseAndExpression](#), [BitwiseOrExpression](#), [RightShiftExpression](#), [LeftShiftExpression](#), [DivExpression](#), [RemainderExpression](#), [MulExpression](#), [SubExpression](#), [AddExpression](#), [AtExpression](#), [MatchesExpression](#), [InExpression](#), [ParenthesesExpression](#), [Enum](#), and [Range](#).

5.30.2.2 `areOperandsValid()`

```
virtual bool Expression::areOperandsValid ( ) const [pure virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implemented in [ForIntExpression](#), [VarListExpression](#), [ForExpression](#), [OfExpression](#), [SetExpression](#), [ThemExpression](#), [PercentQuantifierExpression](#), [QuantifierExpression](#), [FunctionCallExpression](#), [StructExpression](#), [ArrayExpression](#), [Symbol](#), [StringPropertyExpression](#), [TerminalExpression](#), [UnaryExpression](#), [BinaryExpression](#), [InExpression](#), [ParenthesesExpression](#), [Enum](#), and [Range](#).

5.30.2.3 `getType()`

```
virtual Expression::Type Expression::getType ( ) const [pure virtual]
```

Returns the data type of expression

Implemented in [ForIntExpression](#), [VarListExpression](#), [ForExpression](#), [OfExpression](#), [SetExpression](#), [ThemExpression](#), [PercentQuantifierExpression](#), [QuantifierExpression](#), [FunctionCallExpression](#), [StructExpression](#), [Symbol](#), [StringMatchLengthExpression](#), [StringOffsetExpression](#), [StringCountExpression](#), [StringExpression](#), [RegexExpression](#), [SizeExpression](#), [LiteralExpression](#), [UnaryBoolExpression](#), [UnaryBitwiseExpression](#), [UnaryArithmeticExpression](#), [DefinedExpression](#), [BinaryBoolExpression](#), [BinaryStrRelationalExpression](#), [BinaryRelationalExpression](#), [BinaryBitwiseExpression](#), [RemainderExpression](#), [BinaryArithmeticExpression](#), [AtExpression](#), [MatchesExpression](#), [InExpression](#), [ParenthesesExpression](#), [Enum](#), and [Range](#).

5.30.2.4 `isArithmeticCompatible()`

```
bool Expression::isArithmeticCompatible (
    const ExpressionPtr & l,
    const ExpressionPtr & r ) [static]
```

Check whether two expressions are arithmetically compatible (if implicit conversion exists of their data types).

5.30.2.5 `isComplete()`

```
virtual bool Expression::isComplete ( ) const [pure virtual]
```

Checks whether expression is complete - if it has all mandatory operands.

Implemented in [ForIntExpression](#), [VarListExpression](#), [ForExpression](#), [OfExpression](#), [SetExpression](#), [ThemExpression](#), [PercentQuantifierExpression](#), [QuantifierExpression](#), [FunctionCallExpression](#), [StructExpression](#), [ArrayExpression](#), [Symbol](#), [StringPropertyExpression](#), [StringCountExpression](#), [StringExpression](#), [RegexExpression](#), [SizeExpression](#), [LiteralExpression](#), [UnaryExpression](#), [BinaryExpression](#), [InExpression](#), [ParenthesesExpression](#), [Enum](#), and [Range](#).

5.30.2.6 `isValid()`

```
virtual bool Expression::isValid (
    std::string_view & msg ) const [pure virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implemented in [PercentQuantifierExpression](#), [FunctionCallExpression](#), [StructExpression](#), [ArrayExpression](#), [Symbol](#), [StringPropertyExpression](#), [TerminalExpression](#), [UnaryBoolExpression](#), [UnaryBitwiseExpression](#), [UnaryArithmeticExpression](#), [DefinedExpression](#), [BinaryBoolExpression](#), [BinaryStrRelationalExpression](#), [BinaryRelationalExpression](#), [BinaryBitwiseExpression](#), [RemainderExpression](#), [BinaryArithmeticExpression](#), [AtExpression](#), [MatchesExpression](#), [InExpression](#), [ParenthesesExpression](#), [Enum](#), [Range](#), [ForIntExpression](#), [VarListExpression](#), [ForExpression](#), [OfExpression](#), [SetExpression](#), [ThemExpression](#), and [QuantifierExpression](#).

5.30.2.7 typeToString()

```
std::string_view Expression::typeToString (
    Expression::Type type ) [static]
```

Provides conversion of [Expression::Type](#) value to readable string.

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

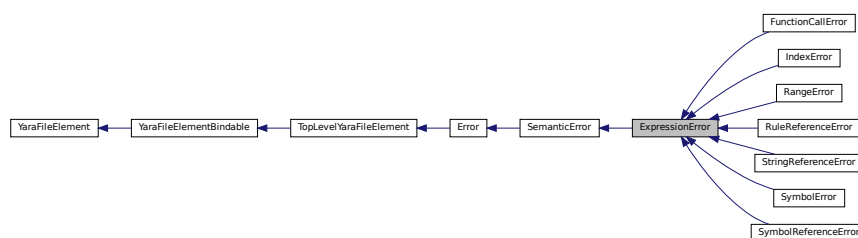
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.31 ExpressionError Class Reference

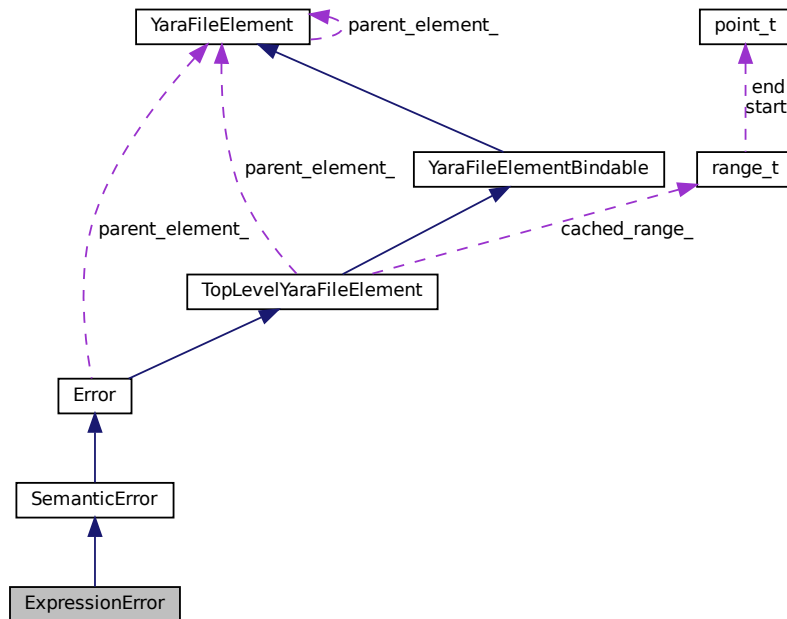
Base class for errors of expression (e. g. type mismatch)

```
#include <error.h>
```

Inheritance diagram for ExpressionError:



Collaboration diagram for ExpressionError:



Public Member Functions

- **ExpressionError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [ExpressionErrorException](#) exception ()

Additional Inherited Members

5.31.1 Detailed Description

Base class for errors of expression (e. g. type mismatch)

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

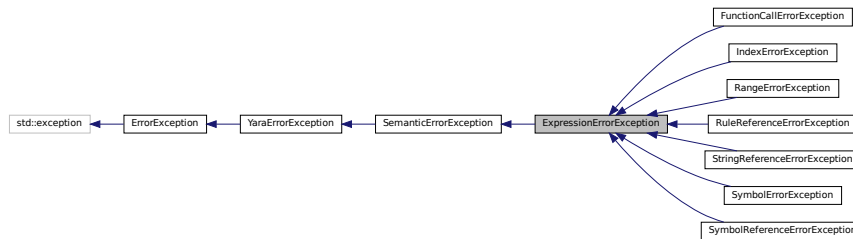
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.32 ExpressionErrorException Class Reference

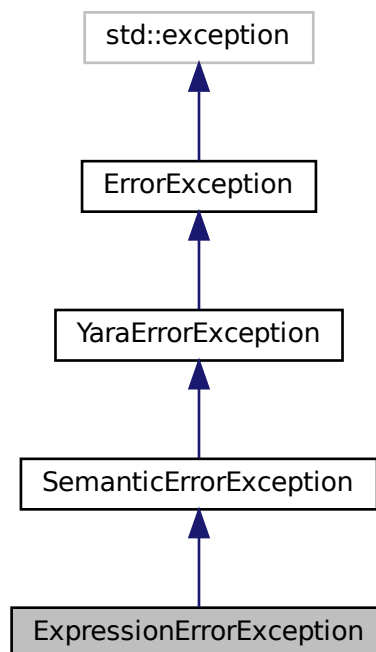
Base class for exception, that represents error, that occurred in expression

```
#include <error.h>
```

Inheritance diagram for ExpressionErrorException:



Collaboration diagram for ExpressionErrorException:



Public Member Functions

- **ExpressionErrorException** (offset_t offset, size_t len, std::string msg={})

5.32.1 Detailed Description

Base class for exception, that represents error, that occurred in expression

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

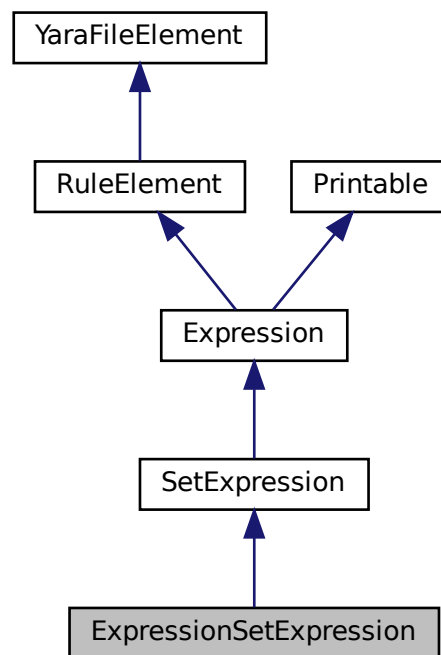
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.33 ExpressionSetExpression Class Reference

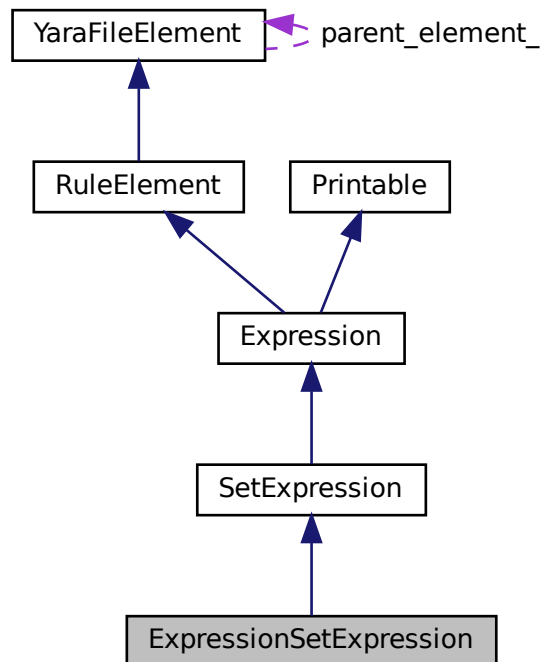
Represents set of expressions.

```
#include <expression.h>
```

Inheritance diagram for ExpressionSetExpression:



Collaboration diagram for ExpressionSetExpression:



Public Member Functions

- **ExpressionSetExpression** (`std::vector< ExpressionPtr > elements`)
- `std::stringstream` [getTextFormatted](#) () const override
- void [accept](#) ([Visitor](#) *v) override

Additional Inherited Members

5.33.1 Detailed Description

Represents set of expressions.

Note

Avast specific object

5.33.2 Member Function Documentation

5.33.2.1 accept()

```
void ExpressionSetExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.33.2.2 getTextFormatted()

```
std::stringstream ExpressionSetExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

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

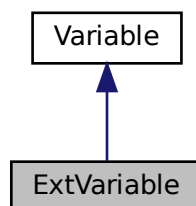
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.34 ExtVariable Class Reference

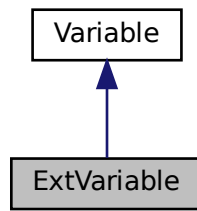
Class that represents external variables.

```
#include <variable.h>
```

Inheritance diagram for ExtVariable:



Collaboration diagram for ExtVariable:



Public Member Functions

- **ExtVariable** (const std::string &id, const std::string &value)
- **ExtVariable** (const std::string &id, int64_t value)
- **ExtVariable** (const std::string &id, bool value)
- const std::unique_ptr< [Literal](#) > & [getValue](#) () const
- void [setValue](#) (std::unique_ptr< [Literal](#) > &&literal)

Additional Inherited Members

5.34.1 Detailed Description

Class that represents external variables.

5.34.2 Member Function Documentation

5.34.2.1 [getValue\(\)](#)

```
const std::unique_ptr< Literal > & ExtVariable::getValue ( ) const
```

Returns [Literal](#) object, that represents value of external variable

5.34.2.2 [setValue\(\)](#)

```
void ExtVariable::setValue (
    std::unique_ptr< Literal > && literal )
```

Sets the value of external variable

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[variable.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[variable.cpp](#)

5.35 FileContext Class Reference

Wrapper for all necessary data to perform incremental reparsing (they are concrete enough)

```
#include <file_context.h>
```

Classes

- struct [cache_entry_t](#)
Entry in stored in cache.
- struct [cache_t](#)
Cache of [FileContext](#) for [offset_t](#) <-> [point_t](#) conversion.

Public Member Functions

- const std::string & [getString](#) () const
- const std::string & [getTempString](#) () const
- void [setString](#) (const std::string &string)
- void [stringEdit](#) (const [offset_t](#) &offset, const [size_t](#) &size, const std::string &new_text)
- void [stringUpdate](#) ()
- void [undoEdits](#) ()
- TSTree * [getTree](#) () const
- void [setTree](#) (TSTree *tree)
- void [clearTree](#) ()
- void [diffTree](#) (TSTree *new_tree, std::vector< [offset_edit_range_t](#) > &diff_ranges)
- void [treeEdit](#) (const TSInputEdit &ts_edit)
- TSInputEdit [toTSEdit](#) (const [offset_t](#) &offset, const [size_t](#) &ins_n, const [size_t](#) &del_n)
- TSInputEdit [toTSEdit](#) (const [offset_t](#) &offset, const [size_t](#) &ins_n, const [size_t](#) &del_n, const [range_t](#) &changed_range)
- [point_t](#) [offsetToPointCached](#) (const [offset_t](#) &offset, const [offset_t](#) &start_offset=0, const [point_t](#) &start_pt={0, 0})
- [offset_t](#) [pointToOffsetCached](#) (const [point_t](#) &point)

5.35.1 Detailed Description

Wrapper for all necessary data to perform incremental reparsing (they are concrete enough)

5.35.2 Member Function Documentation

5.35.2.1 clearTree()

```
void FileContext::clearTree ( )
```

Clears the internal structure with syntax tree

5.35.2.2 diffTree()

```
void FileContext::diffTree (
    TSTree * new_tree,
    std::vector< offset_edit_range_t > & diff_ranges )
```

Compares stored syntax tree and new syntax tree (specified by the first argument), modified ranges found by tree-sitter are appended to the provided vector (second argument)

< Get modified ranges report by the tree sitter

< Convert TSTRange to `offset_edit_range_t` struct

< Free the original TS structure

5.35.2.3 getString()

```
const std::string & FileContext::getString ( ) const
```

Returns reference to internal buffer of parsing context

5.35.2.4 getTempString()

```
const std::string & FileContext::getTempString ( ) const
```

Returns reference to internal buffer of parsing context with temporary (edited) content of the file

5.35.2.5 getTree()

```
TSTree * FileContext::getTree ( ) const
```

Returns internal pointer to structure with syntax tree

5.35.2.6 setString()

```
void FileContext::setString (
    const std::string & string )
```

Sets the new content of the internal buffer < Invalidate cache because string is changed

5.35.2.7 setTree()

```
void FileContext::setTree (
    TSTree * tree )
```

Sets the internal ptr to internal structure with syntax tree to the new ptr

5.35.2.8 stringEdit()

```
void FileContext::stringEdit (
    const offset_t & offset,
    const size_t & size,
    const std::string & new_text )
```

Partly updates temporary internal string buffer < Update the part of tmp internal buffer with string

5.35.2.9 stringUpdate()

```
void FileContext::stringUpdate ( )
```

Performs update of internal buffer, all edits since last [FileContext::stringUpdate](#) call are saved into internal buffer < Invalidate cache because string is changed

5.35.2.10 toTSEdit() [1/2]

```
TSInputEdit FileContext::toTSEdit (
    const offset_t & offset,
    const size_t & ins_n,
    const size_t & del_n )
```

Performs edit of parsing context < Convert edit params to TSInputEdit for tree-sitter

5.35.2.11 toTSEdit() [2/2]

```
TSInputEdit FileContext::toTSEdit (
    const offset_t & offset,
    const size_t & ins_n,
    const size_t & del_n,
    const range_t & changed_range )
```

< Convert edit params to TSInputEdit for tree-sitter

5.35.2.12 treeEdit()

```
void FileContext::treeEdit (
    const TSInputEdit & ts_edit )
```

Performs edit above the TS tree structure

5.35.2.13 undoEdits()

```
void FileContext::undoEdits ( )
```

Discards string edits, that were done since last [FileContext::stringUpdate](#) call

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

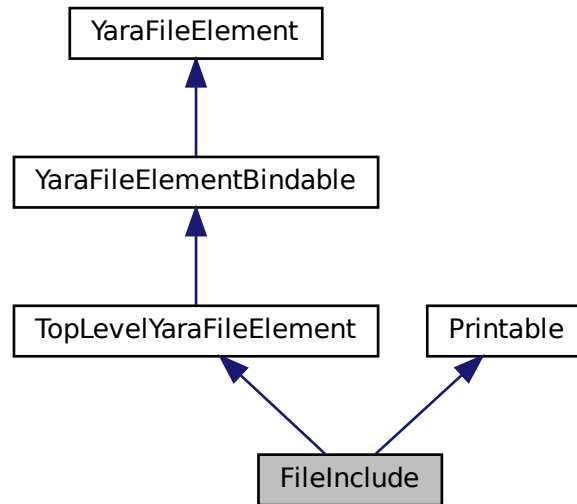
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/file_context.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/file_context.cpp](#)

5.36 FileInclude Class Reference

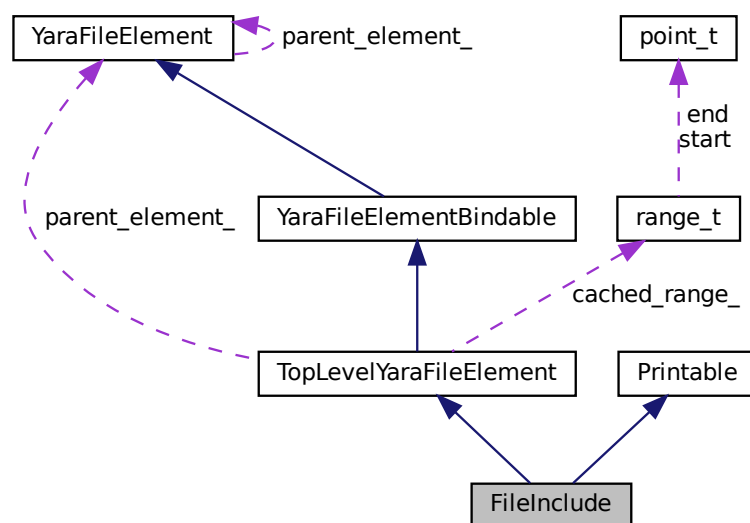
Class representing "include <path>" structure in file.

```
#include <other.h>
```

Inheritance diagram for FileInclude:



Collaboration diagram for FileInclude:



Public Member Functions

- `const std::string & getId () const`
- `void setPath (const std::string &path, const std::string &base_path)`
- `const std::string & getPath () const`
- `void setFile (const std::shared_ptr< YaraFile > &yara_file)`
- `std::shared_ptr< YaraFile > getFile () const`
- `std::stringstream getTextFormatted () const override`

Additional Inherited Members

5.36.1 Detailed Description

Class representing "include <path>" structure in file.

5.36.2 Member Function Documentation

5.36.2.1 getFile()

```
std::shared_ptr< YaraFile > FileInclude::getFile ( ) const
```

Returns pointer to included [YaraFile](#)

5.36.2.2 getId()

```
const std::string & FileInclude::getId ( ) const
```

Returns id of file include - the id is in this case same as path

5.36.2.3 getPath()

```
const std::string & FileInclude::getPath ( ) const
```

Returns path of included file

5.36.2.4 getTextFormatted()

```
std::stringstream FileInclude::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

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

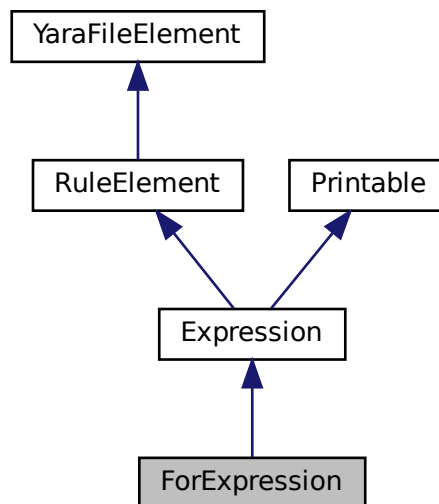
- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/other.h`
- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/other.cpp`

5.37 ForExpression Class Reference

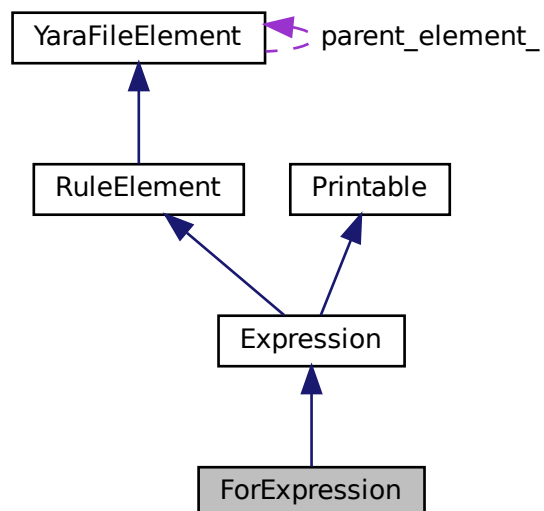
Represents 'for' operator that have rule set as iterable set.

```
#include <expression.h>
```

Inheritance diagram for ForExpression:



Collaboration diagram for ForExpression:



Public Member Functions

- **ForExpression** (ExpressionPtr quantifier, ExpressionPtr set, ExpressionPtr inner_expr)
- bool [isValid](#) (std::string_view &) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- [Expression::Type](#) [getType](#) () const override
- void [accept](#) (Visitor *v) override
- const ExpressionPtr & [getQuantifier](#) () const
- const ExpressionPtr & [getSet](#) () const
- const ExpressionPtr & [getExpression](#) () const

Additional Inherited Members

5.37.1 Detailed Description

Represents 'for' operator that have rule set as iterable set.

It is used for applying the same condition to many strings

Note

Syntax structure in YARA : for <quantifier> of <string_set> : (<boolean_expression>)

5.37.2 Member Function Documentation

5.37.2.1 [accept\(\)](#)

```
void ForExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.37.2.2 [areOperandsValid\(\)](#)

```
bool ForExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.37.2.3 getTextFormatted()

```
std::stringstream ForExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.37.2.4 getType()

```
Expression::Type ForExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.37.2.5 isComplete()

```
bool ForExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.37.2.6 isValid()

```
bool ForExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

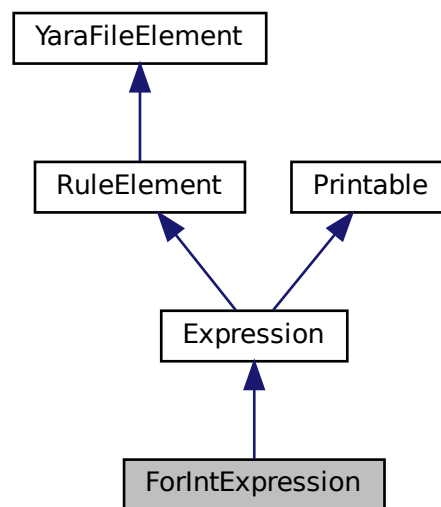
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.38 ForIntExpression Class Reference

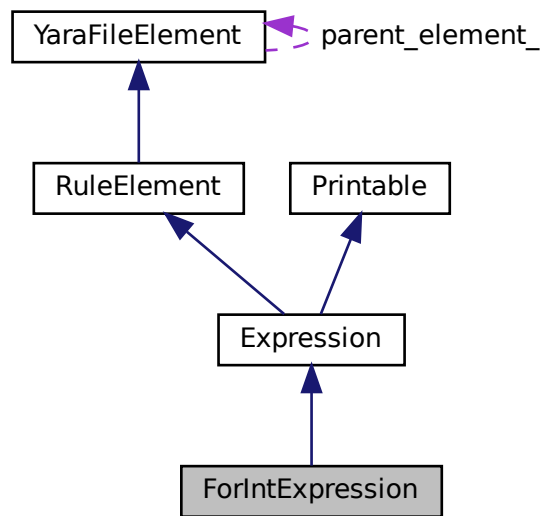
Represents 'for' operator with defined variables for inner expression and with any iterable.

```
#include <expression.h>
```

Inheritance diagram for ForIntExpression:



Collaboration diagram for ForIntExpression:



Public Member Functions

- **ForIntExpression** (ExpressionPtr quantifier, std::shared_ptr< [VarListExpression](#) > list, ExpressionPtr iterable, ExpressionPtr expr)
- bool [isValid](#) (std::string_view &) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- [Expression::Type](#) [getType](#) () const override
- void [accept](#) (Visitor *v) override
- const ExpressionPtr & [getQuantifier](#) () const
- const std::shared_ptr< [VarListExpression](#) > & [getVarList](#) () const
- const ExpressionPtr & [getIterable](#) () const
- const ExpressionPtr & [getExpression](#) () const

Additional Inherited Members

5.38.1 Detailed Description

Represents 'for' operator with defined variables for inner expression and with any iterable.

There is another type of 'for' expression - [ForExpression](#) , that have quite different semantics

Note

Syntax structure in YARA : for <quantifier> <var_list> in <iterable> : (<inner_expression>)

5.38.2 Member Function Documentation

5.38.2.1 accept()

```
void ForIntExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.38.2.2 areOperandsValid()

```
bool ForIntExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.38.2.3 getTextFormatted()

```
std::stringstream ForIntExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.38.2.4 getType()

```
Expression::Type ForIntExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.38.2.5 isComplete()

```
bool ForIntExpression::isComplete ( ) const [override], [virtual]
```

Checks whether expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.38.2.6 isValid()

```
bool ForIntExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

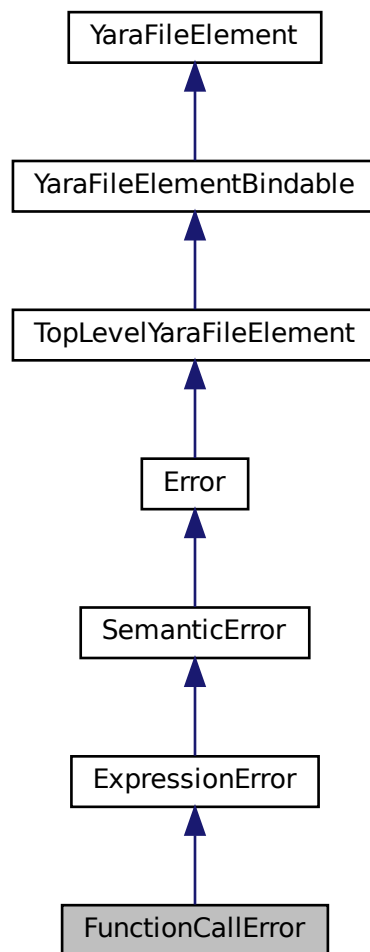
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.39 FunctionCallError Class Reference

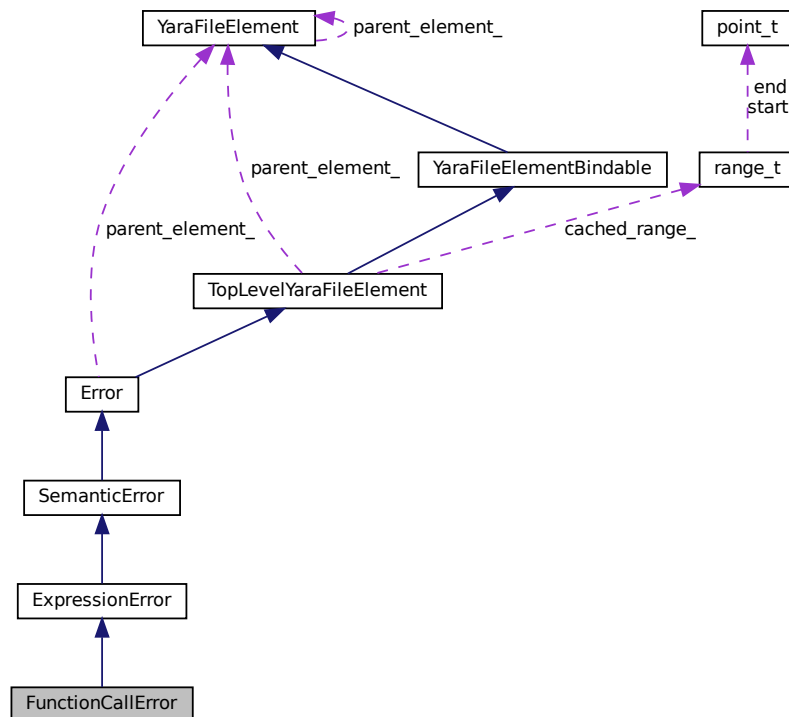
Bad function call (e. g. bad types or bad count of arguments)

```
#include <error.h>
```

Inheritance diagram for FunctionCallError:



Collaboration diagram for FunctionCallError:



Public Member Functions

- **FunctionCallError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [FunctionCallErrorException](#) exception ()

Additional Inherited Members

5.39.1 Detailed Description

Bad function call (e. g. bad types or bad count of arguments)

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

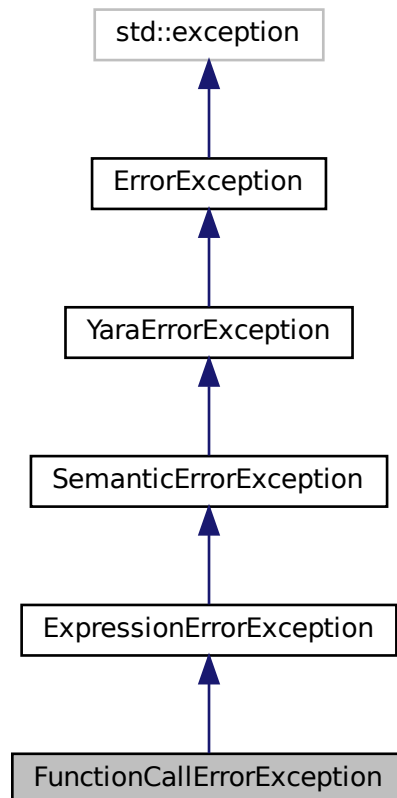
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.40 FunctionCallErrorException Class Reference

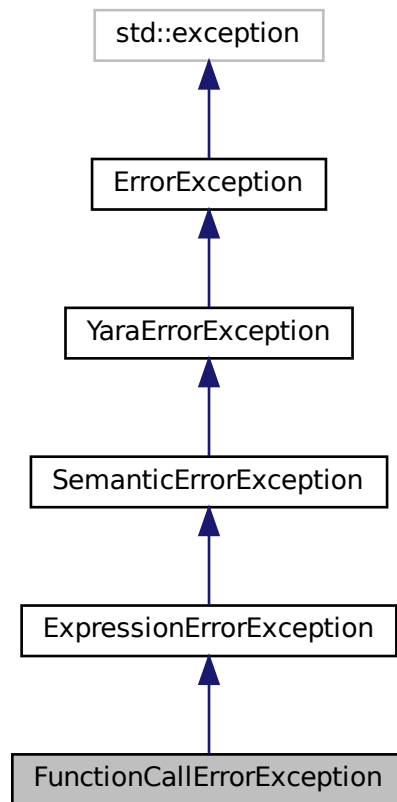
Exception for errors, that occur in function call (e. g. bad types of arguments)

```
#include <error.h>
```

Inheritance diagram for FunctionCallErrorException:



Collaboration diagram for FunctionCallErrorException:



Public Member Functions

- **FunctionCallErrorException** (offset_t offset, size_t len, std::string msg={})

5.40.1 Detailed Description

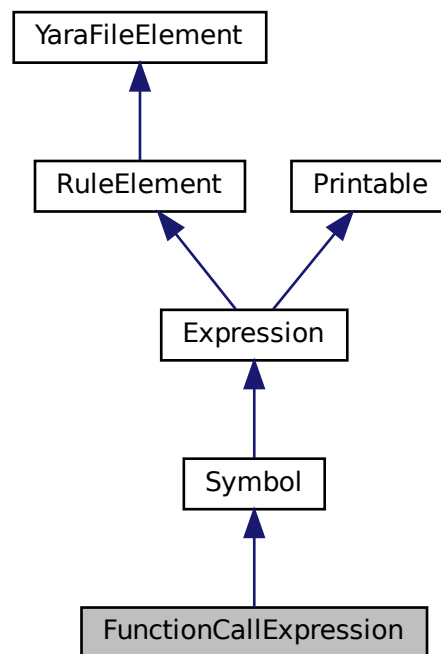
Exception for errors, that occur in function call (e. g. bad types of arguments)

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

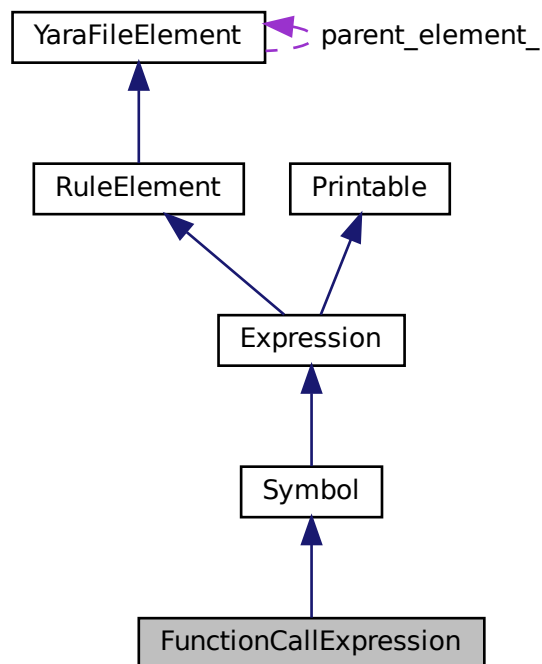
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.41 FunctionCallExpression Class Reference

Inheritance diagram for FunctionCallExpression:



Collaboration diagram for FunctionCallExpression:



Public Member Functions

- **FunctionCallExpression** (std::shared_ptr< [Symbol](#) > id, std::vector< ExpressionPtr > args)
- bool [isComplete](#) () const override
- bool [areOperandsValid](#) () const override
- bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- bool [hasValidArgs](#) () const
- void [accept](#) (Visitor *v) override
- const std::shared_ptr< [Symbol](#) > & [getFunction](#) () const
- const std::vector< ExpressionPtr > & [getArgs](#) () const

Additional Inherited Members

5.41.1 Member Function Documentation

5.41.1.1 accept()

```
void FunctionCallExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.41.1.2 areOperandsValid()

```
bool FunctionCallExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Reimplemented from [Symbol](#).

5.41.1.3 getArgs()

```
const std::vector< ExpressionPtr > & FunctionCallExpression::getArgs ( ) const
```

Returns vector with arguments of function (pointer to expressions)

Note

Returned vector of arguments can be invalid (use [FunctionCallExpression::hasValidArgs](#) to check them them)

5.41.1.4 getTextFormatted()

```
std::stringstream FunctionCallExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Reimplemented from [Symbol](#).

5.41.1.5 getType()

```
Expression::Type FunctionCallExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Reimplemented from [Symbol](#).

5.41.1.6 hasValidArgs()

```
bool FunctionCallExpression::hasValidArgs ( ) const
```

Performs check of arguments of the function

Returns

true if arguments have correct data types and there is correct number of arguments, otherwise false

< The check is done against JSON context of the function

< Field with overloads must have array type

< Field with arguments must have also array type in JSON

< (Expected) Argument iterator

< Real arguments iterator

< Comparison of real argument and expected argument

< Comparison of all real and expected arguments was successful

< There is no other overload of the function

5.41.1.7 isComplete()

```
bool FunctionCallExpression::isComplete ( ) const [override], [virtual]
```

Checks whether expression is complete - if it has all mandatory operands.

Reimplemented from [Symbol](#).

5.41.1.8 isValid()

```
bool FunctionCallExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

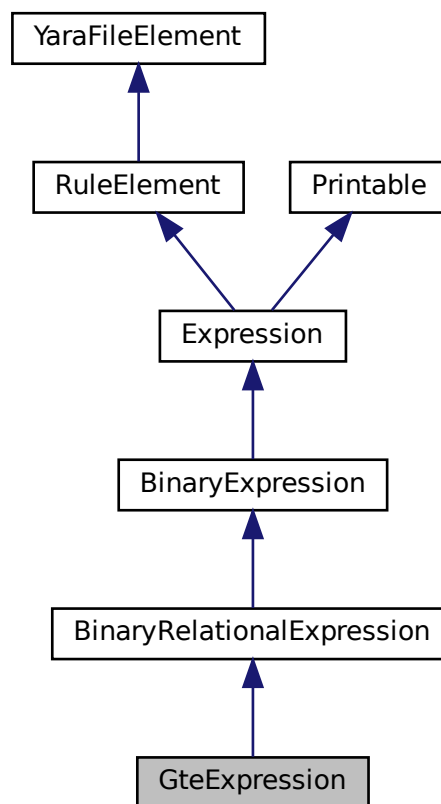
Reimplemented from [Symbol](#).

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

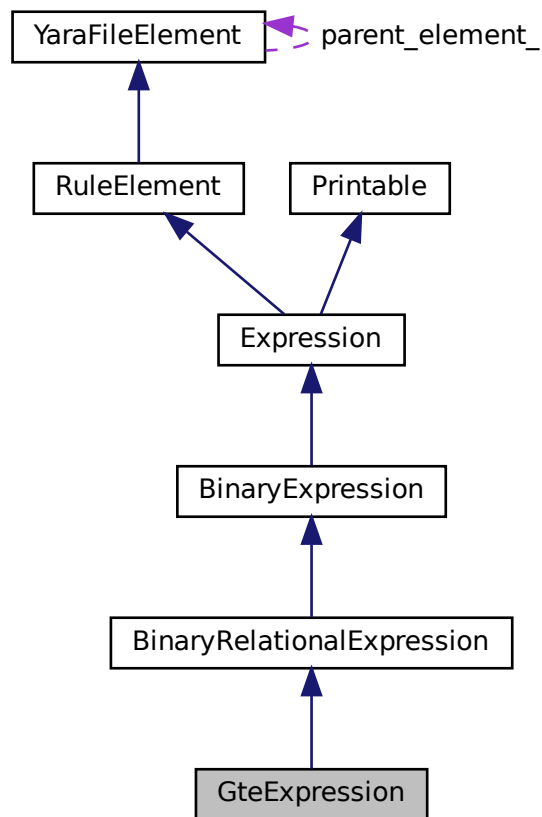
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.42 GteExpression Class Reference

Inheritance diagram for GteExpression:



Collaboration diagram for GteExpression:



Public Member Functions

- **GteExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.42.1 Member Function Documentation

5.42.1.1 [accept\(\)](#)

```
void GteExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.42.1.2 opsign()

```
std::string GtExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

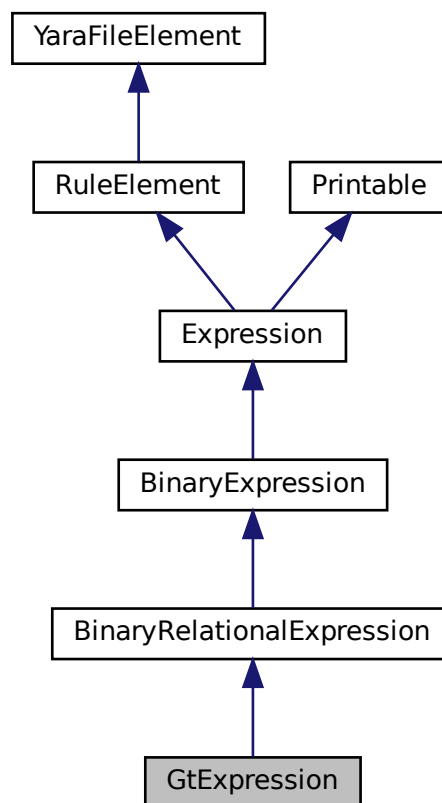
Implements [BinaryExpression](#).

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

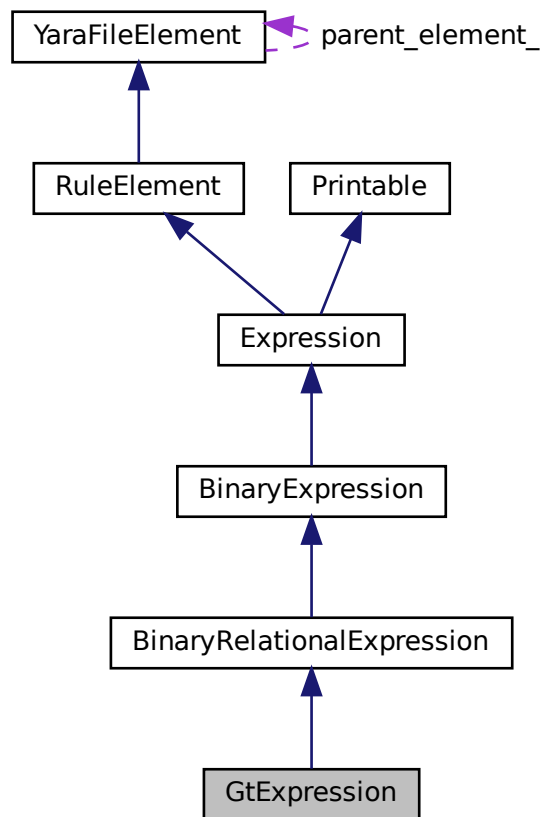
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.43 GtExpression Class Reference

Inheritance diagram for GtExpression:



Collaboration diagram for GtExpression:



Public Member Functions

- **GtExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.43.1 Member Function Documentation

5.43.1.1 [accept\(\)](#)

```
void GtExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.43.1.2 opsign()

```
std::string GtExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

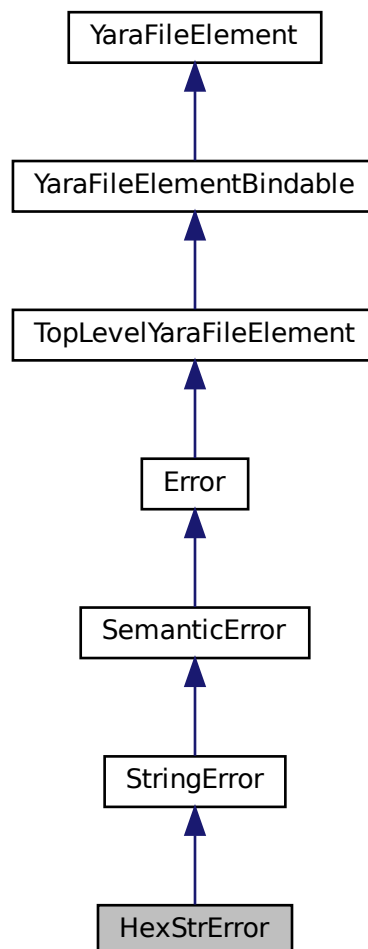
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.44 HexStrError Class Reference

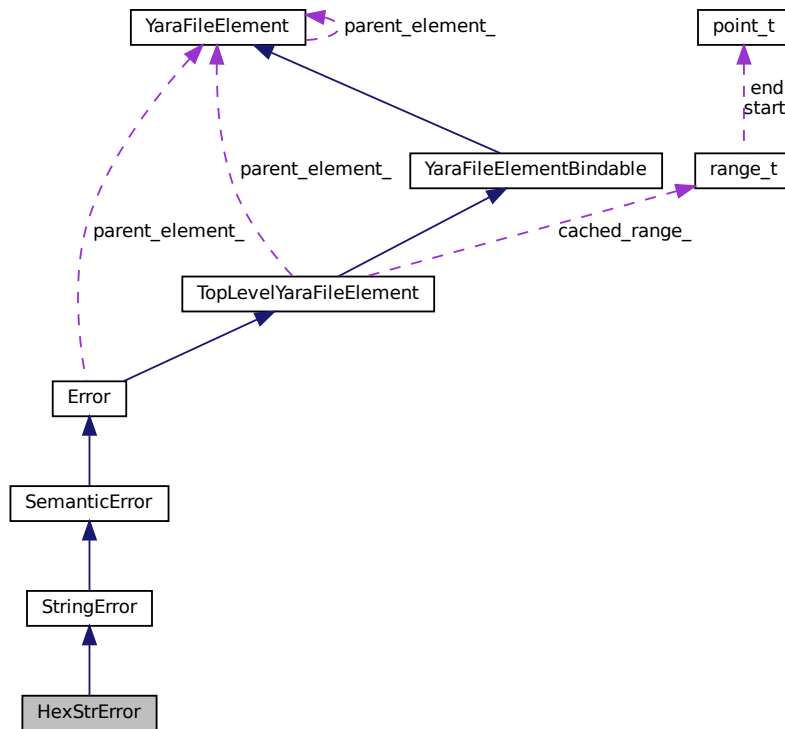
[Error](#) in hexadecimal string (e. g. bad jump boundaries)

```
#include <error.h>
```

Inheritance diagram for HexStrError:



Collaboration diagram for HexStrError:



Public Member Functions

- **HexStrError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [HexStrErrorException](#) exception ()

Additional Inherited Members

5.44.1 Detailed Description

[Error](#) in hexadecimal string (e. g. bad jump boundaries)

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

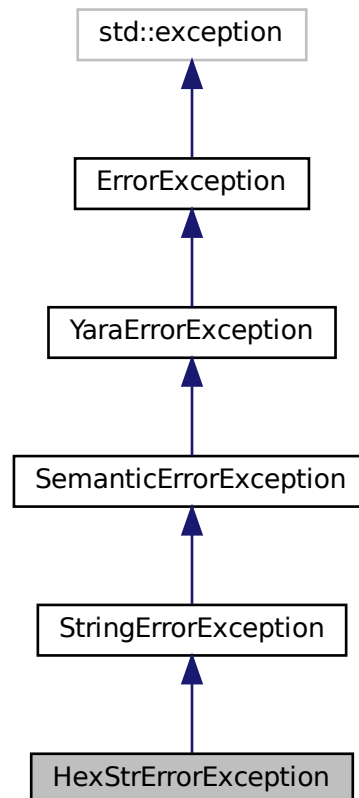
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.45 HexStrErrorException Class Reference

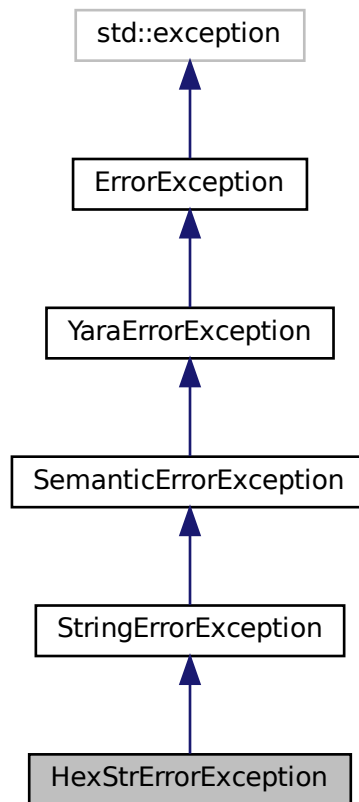
Exception for semantic errors of hexadecimal string (e. g. bad jump interval)

```
#include <error.h>
```

Inheritance diagram for HexStrErrorException:



Collaboration diagram for HexStrErrorException:



Public Member Functions

- **HexStrErrorException** (offset_t offset, size_t len, std::string msg={})

5.45.1 Detailed Description

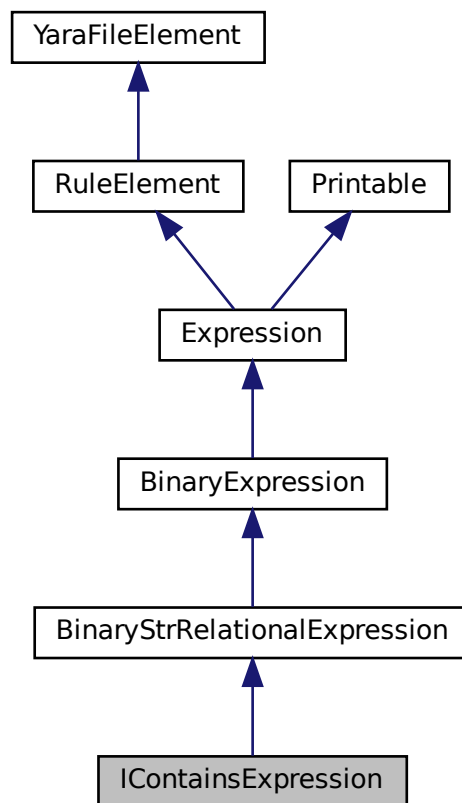
Exception for semantic errors of hexadecimal string (e. g. bad jump interval)

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

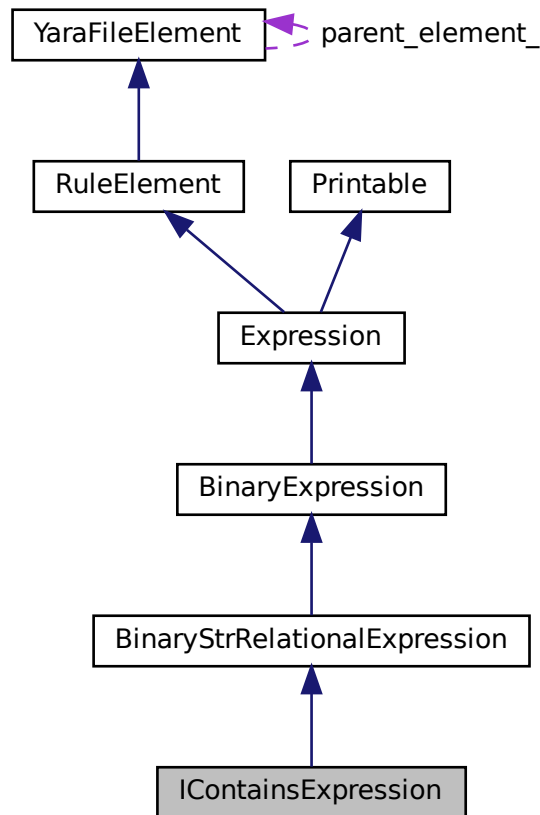
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.46 IContainsExpression Class Reference

Inheritance diagram for IContainsExpression:



Collaboration diagram for IContainsExpression:



Public Member Functions

- **IContainsExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.46.1 Member Function Documentation

5.46.1.1 `accept()`

```
void IContainsExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.46.1.2 opsign()

```
std::string IContainsExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

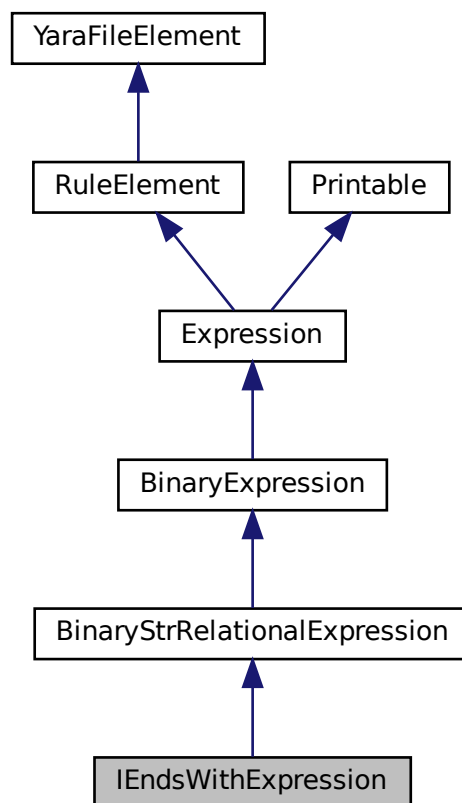
Implements [BinaryExpression](#).

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

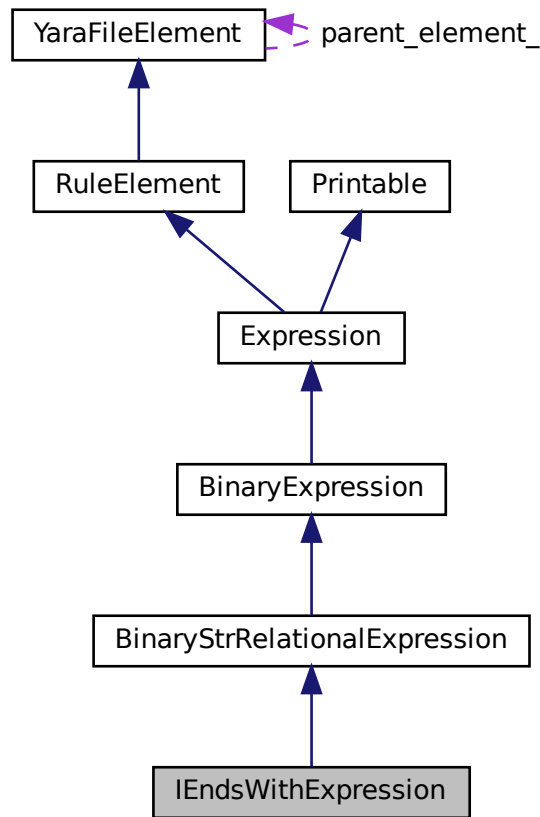
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.47 IEndsWithExpression Class Reference

Inheritance diagram for IEndsWithExpression:



Collaboration diagram for IEndsWithExpression:



Public Member Functions

- **IEndsWithExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.47.1 Member Function Documentation

5.47.1.1 `accept()`

```
void IEndsWithExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.47.1.2 opsign()

```
std::string IEndsWithExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

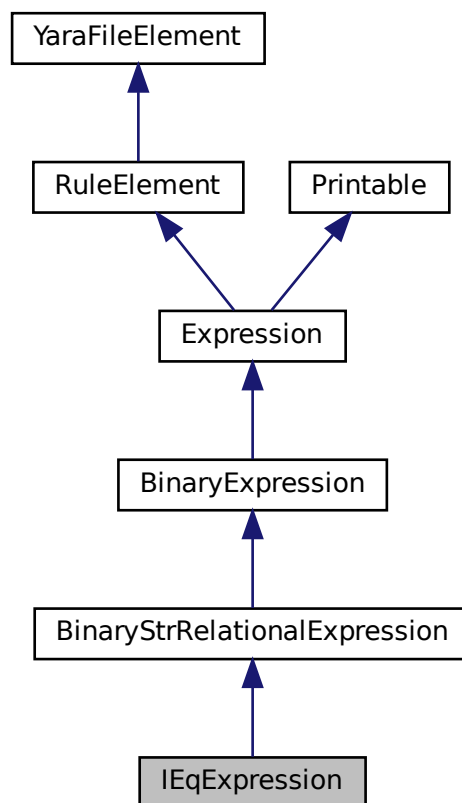
Implements [BinaryExpression](#).

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

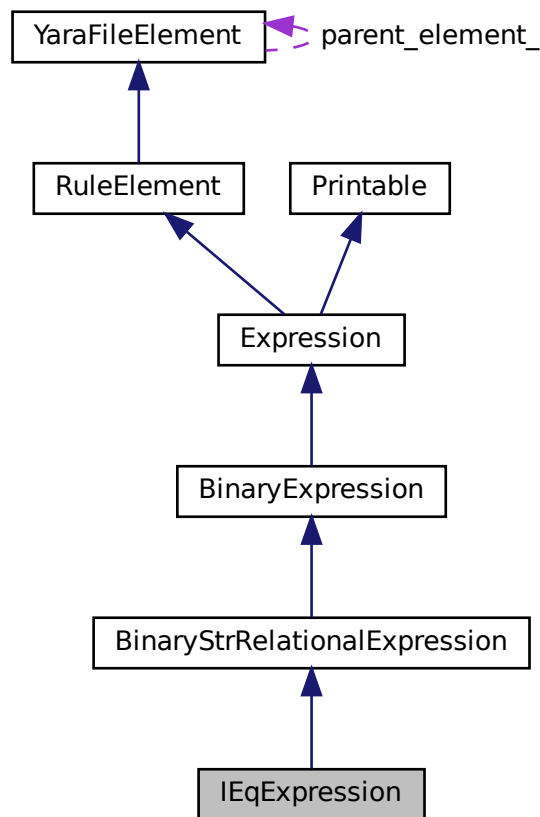
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.48 IEqExpression Class Reference

Inheritance diagram for IEqExpression:



Collaboration diagram for IEqExpression:



Public Member Functions

- **IEqExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.48.1 Member Function Documentation

5.48.1.1 `accept()`

```
void IEqExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.48.1.2 opsign()

```
std::string IEqExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

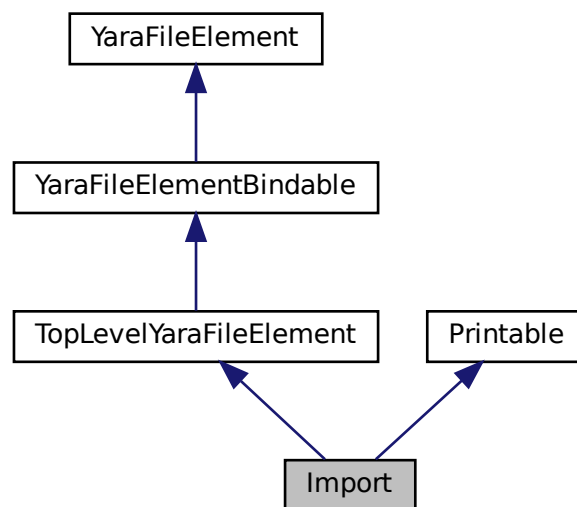
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.49 Import Class Reference

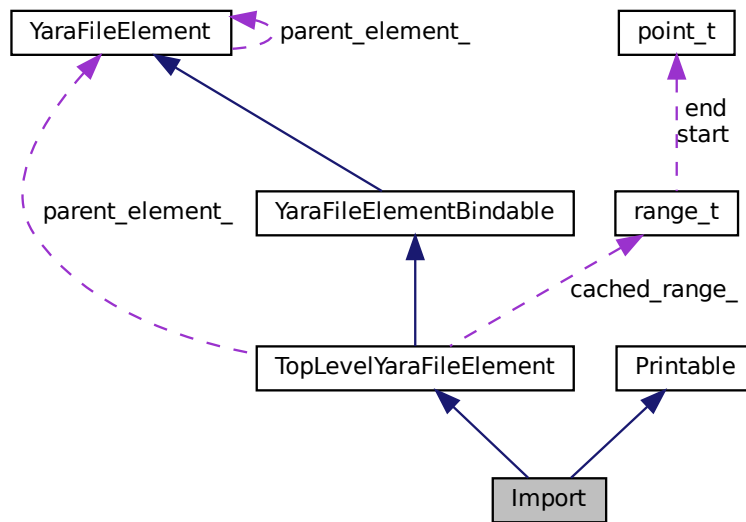
Class representing "import <module>" structure in file.

```
#include <other.h>
```

Inheritance diagram for Import:



Collaboration diagram for Import:



Public Member Functions

- `const std::string & getId () const`
- `void setModuleName (const std::string &module_name)`
- `const std::string & getModuleName () const`
- `void setModule (const std::shared_ptr< Module > &module)`
- `std::shared_ptr< Module > getModule () const`
- `std::stringstream getTextFormatted () const override`

Additional Inherited Members

5.49.1 Detailed Description

Class representing "import <module>" structure in file.

5.49.2 Member Function Documentation

5.49.2.1 getId()

```
const std::string & Import::getId ( ) const
```

Returns id of import - the id is in this case same as if of imported module

5.49.2.2 getTextFormatted()

```
std::stringstream Import::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

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

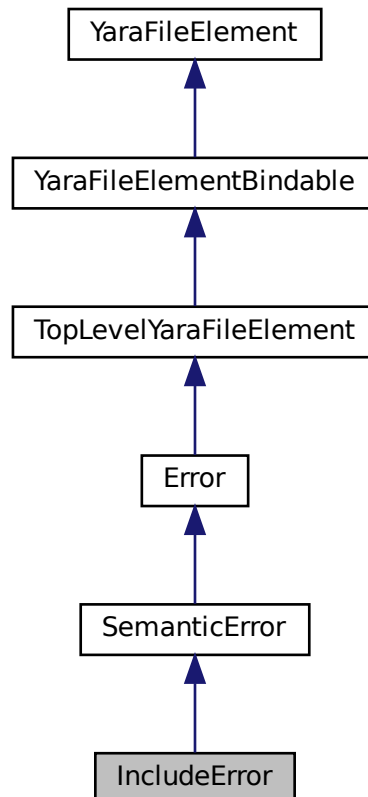
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[other.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[other.cpp](#)

5.50 IncludeError Class Reference

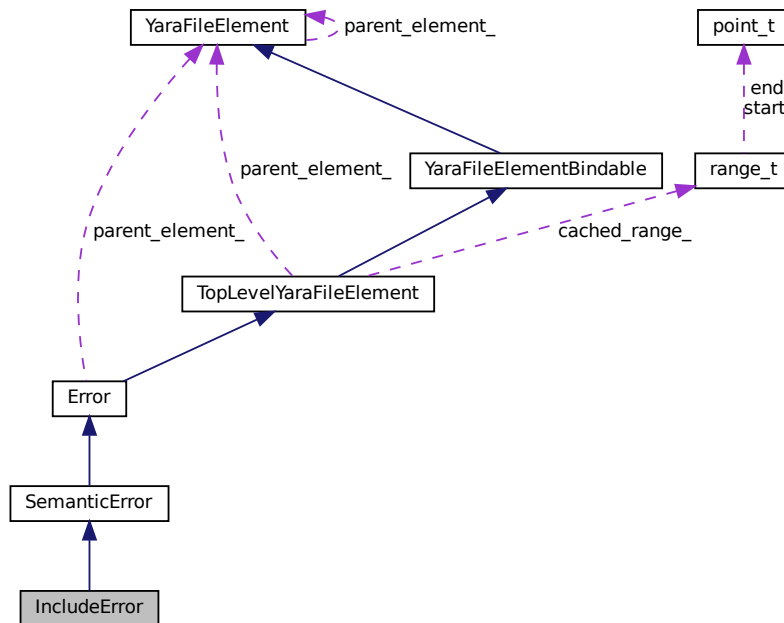
Bad file include (e. g. included file does not exists)

```
#include <error.h>
```

Inheritance diagram for IncludeError:



Collaboration diagram for IncludeError:



Public Member Functions

- **IncludeError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [IncludeErrorException](#) **exception** ()

Additional Inherited Members

5.50.1 Detailed Description

Bad file include (e. g. included file does not exists)

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

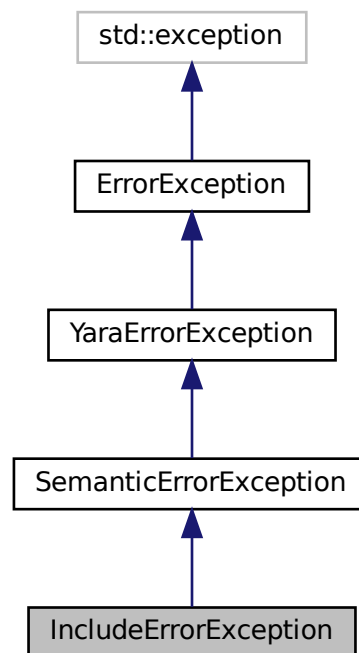
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.51 IncludeErrorException Class Reference

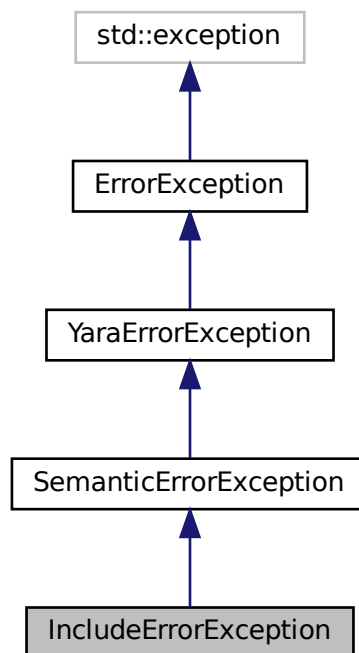
Exception for bad file include (e. g. file is included twice)

```
#include <error.h>
```

Inheritance diagram for IncludeErrorException:



Collaboration diagram for IncludeErrorException:



Public Member Functions

- **IncludeErrorException** (offset_t offset, size_t len, std::string msg={})

5.51.1 Detailed Description

Exception for bad file include (e. g. file is included twice)

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

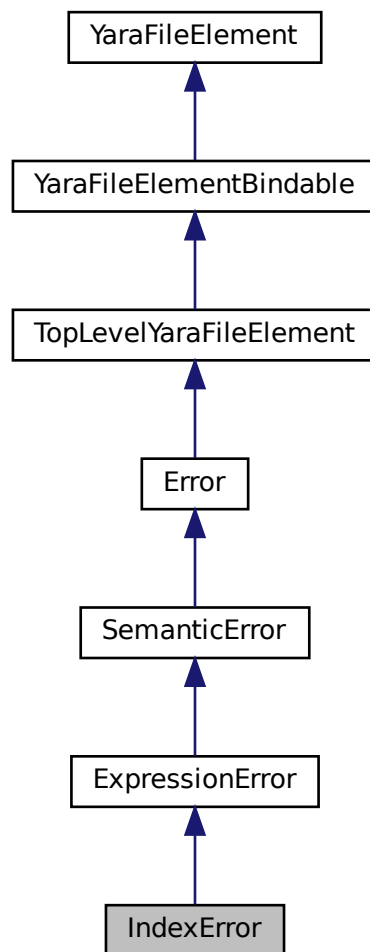
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.52 IndexError Class Reference

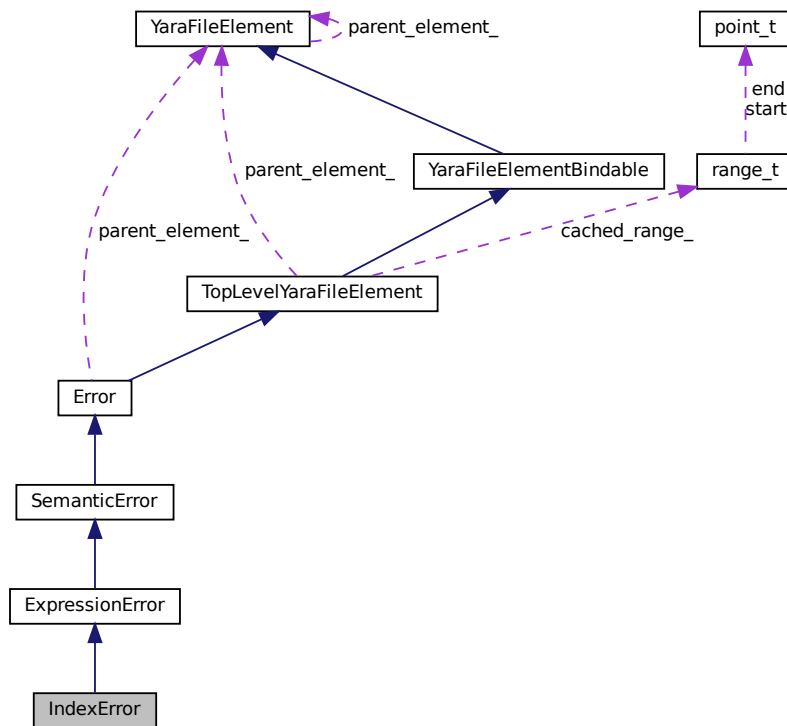
[Error](#) of index in array access expression.

```
#include <error.h>
```

Inheritance diagram for IndexError:



Collaboration diagram for IndexError:



Public Member Functions

- **IndexError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [IndexErrorException](#) **exception** ()

Additional Inherited Members

5.52.1 Detailed Description

[Error](#) of index in array access expression.

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

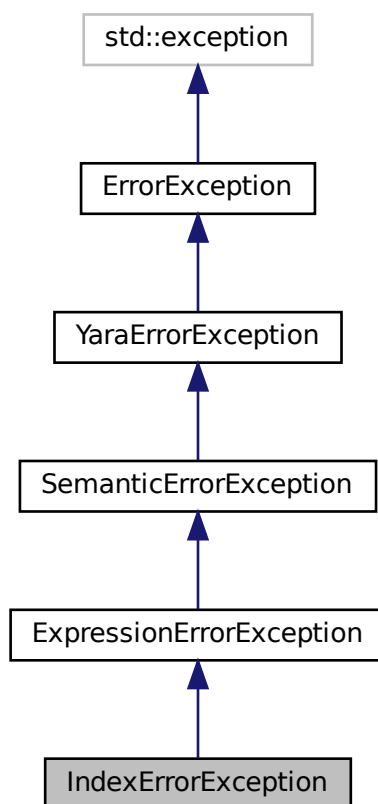
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.53 IndexErrorException Class Reference

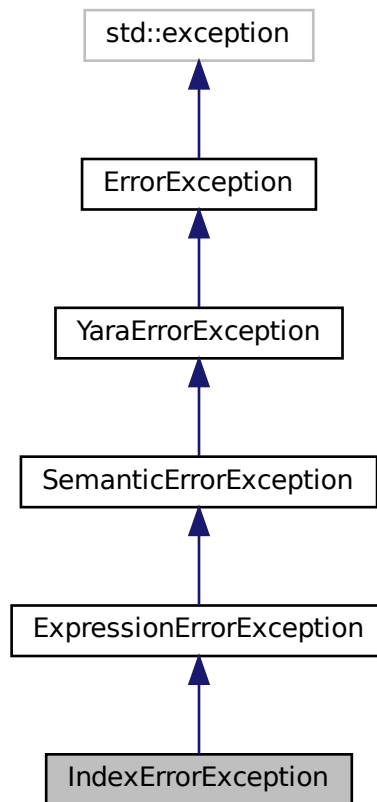
Exception for bad type of index expression in array access expression

```
#include <error.h>
```

Inheritance diagram for IndexErrorException:



Collaboration diagram for IndexErrorException:



Public Member Functions

- **IndexErrorException** (offset_t offset, size_t len, std::string msg={})

5.53.1 Detailed Description

Exception for bad type of index expression in array access expression

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

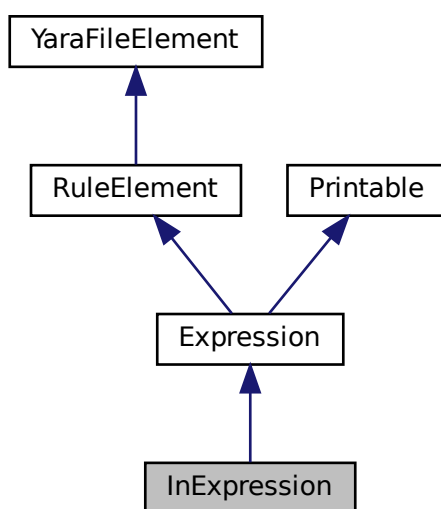
- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h`

5.54 InExpression Class Reference

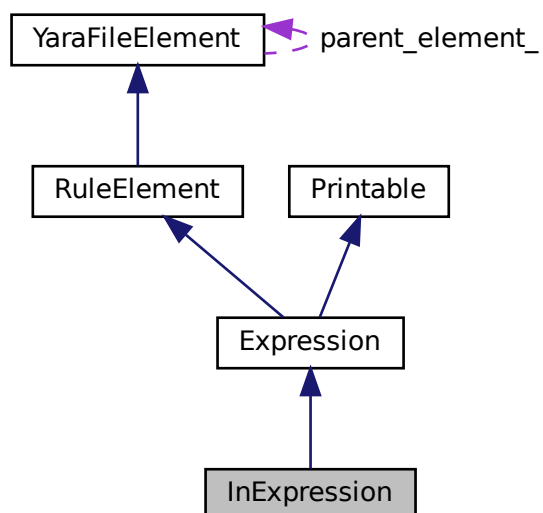
Class representing 'in' operator.

```
#include <expression.h>
```

Inheritance diagram for InExpression:



Collaboration diagram for InExpression:



Public Member Functions

- **InExpression** (ExpressionPtr el, std::shared_ptr< [Range](#) > range)
- bool [isComplete](#) () const override
- bool [areOperandsValid](#) () const override
- bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- std::string [opsign](#) () const
- void [accept](#) ([Visitor](#) *v) override
- const ExpressionPtr & [getElementExpression](#) () const
- const std::shared_ptr< [Range](#) > & [getRangeExpression](#) () const

Additional Inherited Members

5.54.1 Detailed Description

Class representing 'in' operator.

5.54.2 Member Function Documentation

5.54.2.1 [accept\(\)](#)

```
void InExpression::accept (  
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.54.2.2 [areOperandsValid\(\)](#)

```
bool InExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.54.2.3 getTextFormatted()

```
std::stringstream InExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.54.2.4 getType()

```
Expression::Type InExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.54.2.5 isComplete()

```
bool InExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.54.2.6 isValid()

```
bool InExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

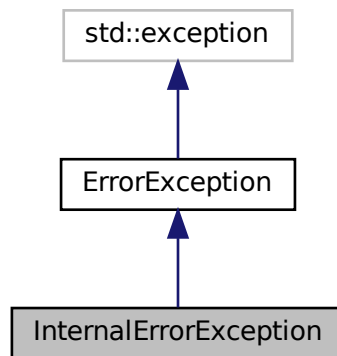
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.55 InternalErrorException Class Reference

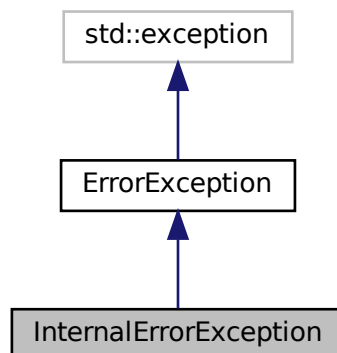
Exception for unexpected errors, that are probably not caused by user (nonrecoverable errors of internal parser etc.)

```
#include <error.h>
```

Inheritance diagram for InternalErrorException:



Collaboration diagram for InternalErrorException:



Public Member Functions

- **InternalErrorException** (std::string msg={})

5.55.1 Detailed Description

Exception for unexpected errors, that are probably not caused by user (nonrecoverable errors of internal parser etc.)

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

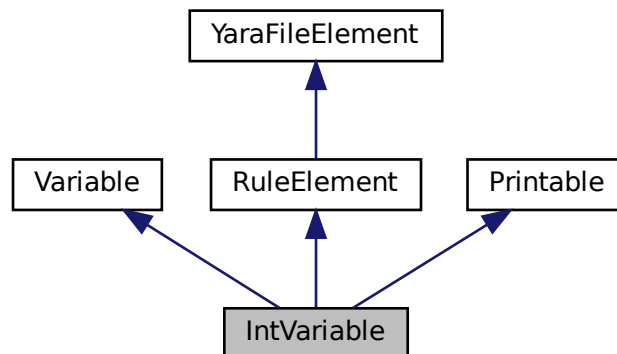
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.56 IntVariable Class Reference

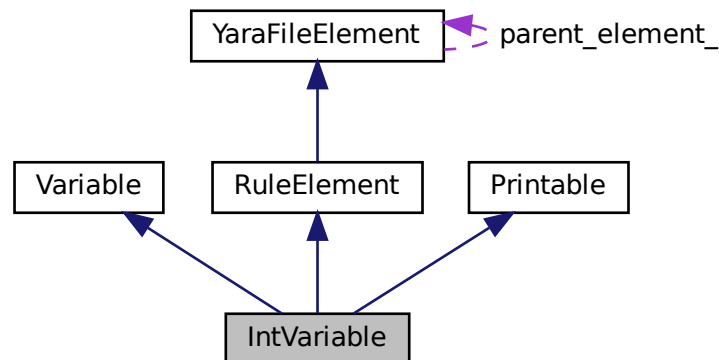
Class, that represents internal variables.

```
#include <variable.h>
```

Inheritance diagram for IntVariable:



Collaboration diagram for IntVariable:



Public Member Functions

- `IntVariable` (const std::string &id)
- const std::shared_ptr< [Expression](#) > & `getValue` () const
- void `setValue` (std::shared_ptr< [Expression](#) > &&expr)
- std::stringstream `getTextFormatted` () const override

Additional Inherited Members

5.56.1 Detailed Description

Class, that represents internal variables.

Note

Internal variables are Avast feature

5.56.2 Member Function Documentation

5.56.2.1 `getTextFormatted()`

```
std::stringstream IntVariable::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.56.2.2 getValue()

```
const std::shared_ptr< Expression > & IntVariable::getValue ( ) const
```

Returns [Expression](#), that represents value of internal variable

5.56.2.3 setValue()

```
void IntVariable::setValue (
    std::shared_ptr< Expression > && expr )
```

Sets the value of internal variable

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

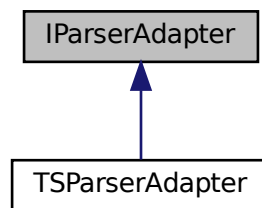
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/variable.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/variable.cpp](#)

5.57 IParserAdapter Class Reference

Interface to which will be connected internal parser by concrete ParserAdapter (the "output socket" of the adapter)

```
#include <parser_adapter.h>
```

Inheritance diagram for IParserAdapter:



Public Member Functions

- **IParserAdapter** (std::shared_ptr< [YaramodConfig](#) > config)
- virtual YaraSourcePtr [createYaraSource](#) (const std::string &string)=0
- virtual YaraSourcePtr **createYaraSource** (const std::string &string, const std::string &entry_file_path)=0
- virtual void [updateYaraSource](#) (const std::string &string, [YaraSource](#) *old_src)=0
- virtual void **updateYaraSource** (const std::string &string, const std::string &entry_file_path, [YaraSource](#) *old_src)=0
- virtual YaraSourcePtr [createYaraSourceFromFile](#) (std::string_view path)=0
- virtual void [updateYaraSourceFromFile](#) (std::string_view path, [YaraSource](#) *old_src)=0

Protected Attributes

- `std::shared_ptr< YaramodConfig > config_`

5.57.1 Detailed Description

Interface to which will be connected internal parser by concrete ParserAdapter (the "output socket" of the adapter)

5.57.2 Member Function Documentation

5.57.2.1 `createYaraSource()`

```
virtual YaraSourcePtr IParserAdapter::createYaraSource (
    const std::string & string ) [pure virtual]
```

Creates [YaraSource](#) object from the given string

Parameters

<i>string</i>	String in YARA language that should be parsed
---------------	---

Returns

Unique ptr to [YaraSource](#)

Implemented in [TSParserAdapter](#).

5.57.2.2 `createYaraSourceFromFile()`

```
virtual YaraSourcePtr IParserAdapter::createYaraSourceFromFile (
    std::string_view path ) [pure virtual]
```

Creates [YaraSource](#) object from the file

Parameters

<i>path</i>	Path leading to file in YARA language that should be parsed
-------------	---

Returns

Unique ptr to [YaraSource](#)

Implemented in [TSParserAdapter](#).

5.57.2.3 updateYaraSource()

```
virtual void IParserAdapter::updateYaraSource (
    const std::string & string,
    YaraSource * old_src ) [pure virtual]
```

Updates [YaraSource](#) object, that were created from the string

Parameters

<i>string</i>	String in YARA language that should be reparsed
<i>old_src</i>	Old YaraSource object, that should be updated

Implemented in [TSParserAdapter](#).

5.57.2.4 updateYaraSourceFromFile()

```
virtual void IParserAdapter::updateYaraSourceFromFile (
    std::string_view path,
    YaraSource * old_src ) [pure virtual]
```

Updates [YaraSource](#) object, that were created from the file

Parameters

<i>path</i>	Path leading to file in YARA language that should be reparsed
<i>old_src</i>	Old YaraSource object, that should be updated

Note

It is not necessary to pass the same path as was passed to [IParserAdapter::createYaraSourceFromFile](#), if it is allowed to change source file, depends on concrete ParserAdapter

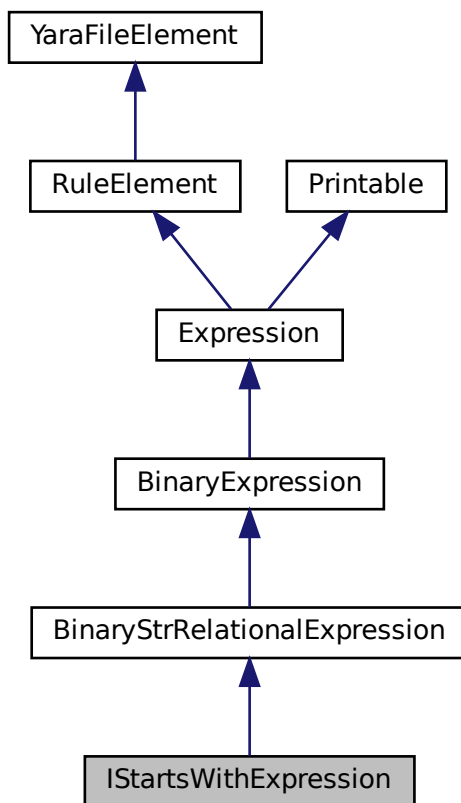
Implemented in [TSParserAdapter](#).

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

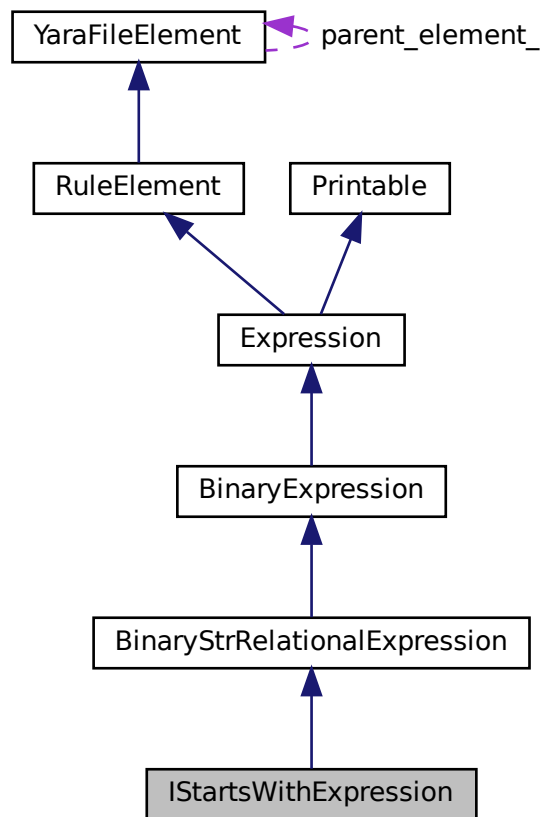
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/parser_adapter.h](#)

5.58 IStartsWithExpression Class Reference

Inheritance diagram for IStartsWithExpression:



Collaboration diagram for IStartsWithExpression:



Public Member Functions

- **IStartsWithExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.58.1 Member Function Documentation

5.58.1.1 `accept()`

```
void IStartsWithExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.58.1.2 opsign()

```
std::string IStartsWithExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.59 RuleModifierContainer::iterator Class Reference

Iterator for [RuleModifierContainer](#).

```
#include <rule.h>
```

Public Types

- using **iterator_category** = std::forward_iterator_tag
- using **value_type** = [RuleModifier](#)
- using **difference_type** = [RuleModifier](#)
- using **pointer** = [RuleModifier](#) *
- using **reference** = [RuleModifier](#) &

Public Member Functions

- **iterator** (const [RuleModifierContainer](#) *container)
- [RuleModifier](#) * **operator*** () const
- [RuleModifier](#) * **operator->** () const
- **iterator** & **operator++** ()
- **iterator** **operator++** (int)

Friends

- class **RuleModifierContainer**
- bool **operator==** (const [iterator](#) &l, const [iterator](#) &r)
- bool **operator!=** (const [iterator](#) &l, const [iterator](#) &r)

5.59.1 Detailed Description

Iterator for [RuleModifierContainer](#).

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

- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/rule.cpp](#)

5.60 StringModifierContainer::iterator Class Reference

Iterator for [StringModifierContainer](#).

```
#include <string.h>
```

Public Types

- using **iterator_category** = std::forward_iterator_tag
- using **value_type** = [StringModifier](#)
- using **difference_type** = [StringModifier](#)
- using **pointer** = [StringModifier](#) *
- using **reference** = [StringModifier](#) &

Public Member Functions

- **iterator** (const [StringModifierContainer](#) *container)
- [StringModifier](#) * **operator*** () const
- [StringModifier](#) * **operator->** () const
- **iterator** & **operator++** ()
- **iterator** **operator++** (int)

Friends

- class **StringModifierContainer**
- bool **operator==** (const [iterator](#) &l, const [iterator](#) &r)
- bool **operator!=** (const [iterator](#) &l, const [iterator](#) &r)

5.60.1 Detailed Description

Iterator for [StringModifierContainer](#).

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[string.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[string.cpp](#)

5.61 SymTab< V >::iterator Class Reference

[SymTab](#) iterator (just a wrapper above [OffsetMap::const_iterator](#))

```
#include <symtab.h>
```

Public Types

- using **iterator_category** = std::forward_iterator_tag
- using **value_type** = V
- using **difference_type** = V
- using **pointer** = V *
- using **reference** = V &

Public Member Functions

- **iterator** (const OffsetMap::const_iterator &it)
- V * **operator*** () const
- V * **operator->** () const
- **iterator** & **operator++** ()
- **iterator** **operator++** (int)

Friends

- class **SymTab**
- bool **operator==** (const [iterator](#) &l, const [iterator](#) &r)
- bool **operator!=** (const [iterator](#) &l, const [iterator](#) &r)

5.61.1 Detailed Description

```
template<class V>
class SymTab< V >::iterator
```

[SymTab](#) iterator (just a wrapper above OffsetMap::const_iterator)

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[symtab.h](#)

5.62 SymTabDuplId< V >::iterator Class Reference

[SymTabDuplId](#) iterator.

```
#include <symtab.h>
```

Public Types

- using **iterator_category** = std::forward_iterator_tag
- using **value_type** = V
- using **difference_type** = V
- using **pointer** = V *
- using **reference** = V &

Public Member Functions

- **iterator** (const OffsetMap::const_iterator &it)
- **V * operator*** () const
- **V * operator->** () const
- **iterator & operator++** ()
- **iterator operator++** (int)

Friends

- class **SymTabDuplId**
- bool **operator==** (const iterator &l, const iterator &r)
- bool **operator!=** (const iterator &l, const iterator &r)

5.62.1 Detailed Description

```
template<class V>
class SymTabDuplId< V >::iterator
```

[SymTabDuplId](#) iterator.

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[symtab.h](#)

5.63 SymTabDuplId< V >::Pool::iterator Class Reference

[Pool](#) iterator.

```
#include <symtab.h>
```

Public Types

- using **iterator_category** = std::forward_iterator_tag
- using **value_type** = V
- using **difference_type** = V
- using **pointer** = V *
- using **reference** = V &

Public Member Functions

- **iterator** (const DataContainer::const_iterator &it)
- **V * operator*** () const
- **V * operator->** () const
- **iterator & operator++** ()
- **iterator operator++** (int)

Friends

- class **Pool**
- bool **operator==** (const [iterator](#) &l, const [iterator](#) &r)
- bool **operator!=** (const [iterator](#) &l, const [iterator](#) &r)

5.63.1 Detailed Description

```
template<class V>
class SymTabDuplId< V >::Pool::iterator
```

[Pool](#) iterator.

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[symtab.h](#)

5.64 SymTabOffset< V >::iterator Class Reference

[SymTabOffset](#) iterator.

```
#include <symtab.h>
```

Public Types

- using **iterator_category** = std::forward_iterator_tag
- using **value_type** = std::unique_ptr< V >
- using **difference_type** = std::unique_ptr< V >
- using **pointer** = std::unique_ptr< V > *
- using **reference** = std::unique_ptr< V > &

Public Member Functions

- **iterator** (const OffsetMap::const_iterator &it)
- V * **operator*** () const
- V * **operator->** () const
- [iterator](#) & **operator++** ()
- [iterator](#) **operator++** (int)

Friends

- class **SymTabOffset**
- bool **operator==** (const [iterator](#) &l, const [iterator](#) &r)
- bool **operator!=** (const [iterator](#) &l, const [iterator](#) &r)

5.64.1 Detailed Description

```
template<class V>  
class SymTabOffset< V >::iterator
```

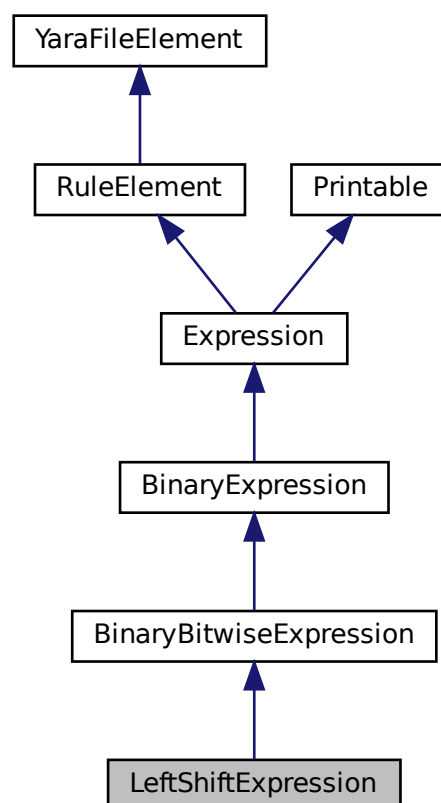
[SymTabOffset](#) iterator.

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

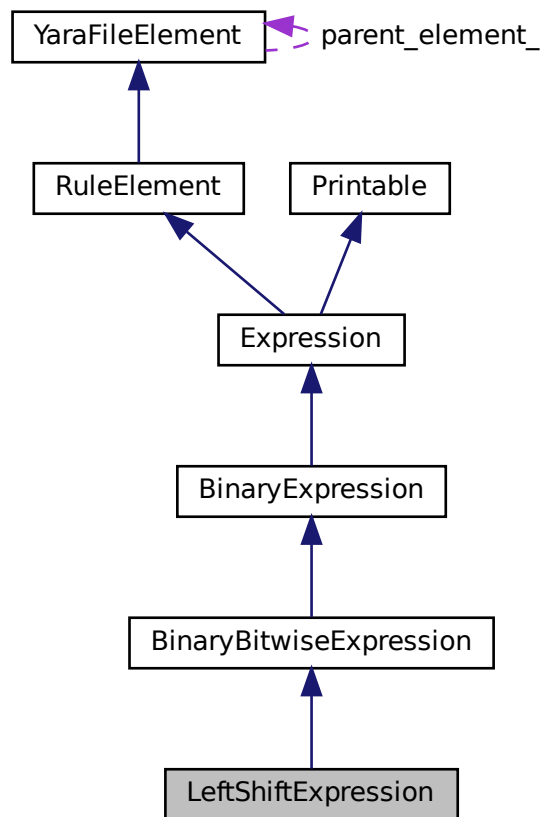
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/symtab.h](#)

5.65 LeftShiftExpression Class Reference

Inheritance diagram for LeftShiftExpression:



Collaboration diagram for LeftShiftExpression:



Public Member Functions

- **LeftShiftExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.65.1 Member Function Documentation

5.65.1.1 [accept\(\)](#)

```
void LeftShiftExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.65.1.2 opsign()

```
std::string LeftShiftExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

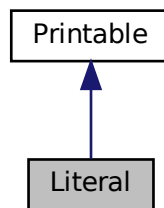
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.66 Literal Class Reference

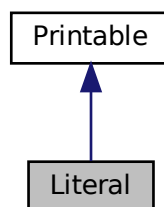
Object that represents literal, that may appear inside a rule definition.

```
#include <literal.h>
```

Inheritance diagram for Literal:



Collaboration diagram for Literal:



Public Member Functions

- `template<typename T >`
Literal (T new_value)
- `template<typename T >`
`void` [set](#) (T new_value)
- `const LiteralValue &` [getRaw](#) () `const`
- `template<typename T >`
`const T &` [get](#) () `const`
- `template<typename T >`
`bool` [isType](#) () `const`
- `bool` [isUndefined](#) () `const`
- `bool` [isString](#) () `const`
- `bool` [isInt](#) () `const`
- `bool` [isFloat](#) () `const`
- `bool` [isBool](#) () `const`
- `std::stringstream` [getTextFormatted](#) () `const`

5.66.1 Detailed Description

Object that represents literal, that may appear inside a rule definition.

5.66.2 Member Function Documentation

5.66.2.1 [get\(\)](#)

```
template<typename T >
const T & Literal::get
```

Returns the concrete value of [Literal](#)

5.66.2.2 [getRaw\(\)](#)

```
const LiteralValue & Literal::getRaw ( ) const
```

Returns the new value of literals (std::get can be used to get concrete value)

5.66.2.3 [getTextFormatted\(\)](#)

```
std::stringstream Literal::getTextFormatted ( ) const [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.66.2.4 isType()

```
template<typename T >  
bool Literal::isType
```

Checks if [Literal](#) contains the value of given type

Returns

true if [Literal](#) contains value of provided data type

5.66.2.5 isUndefined()

```
bool Literal::isUndefined ( ) const
```

Checks if [Literal](#) has undefined value (same as `getType<std::monostate>`)

5.66.2.6 set()

```
template<typename T >  
void Literal::set (  
    T new_value )
```

Sets the value of literal

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

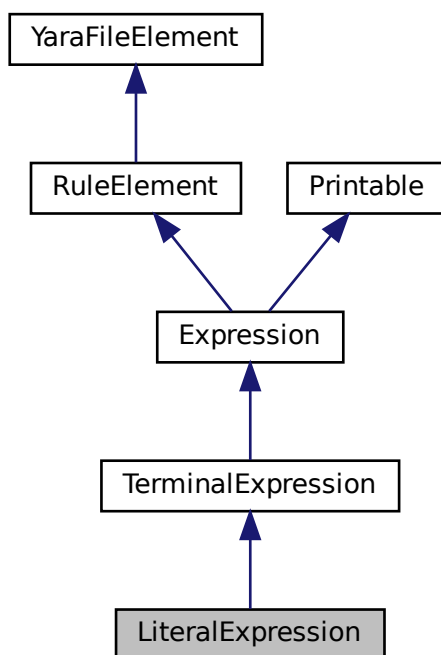
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/literal.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/literal.cpp](#)

5.67 LiteralExpression Class Reference

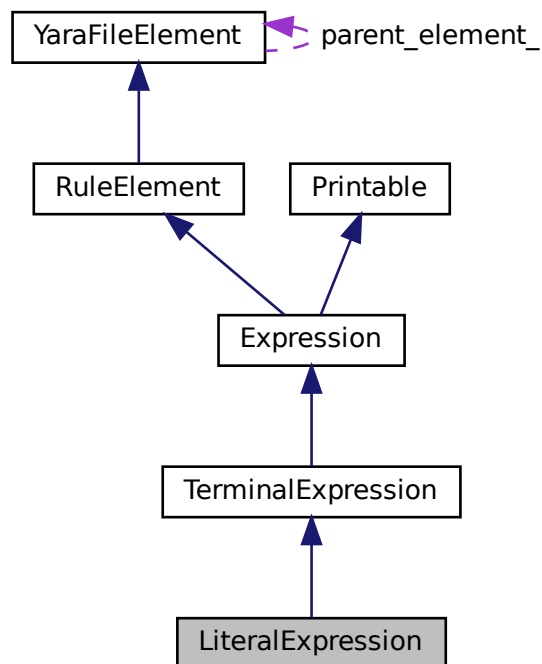
Represents [Literal](#) in expression. Possible terminal node of expression tree (it does not have any other [Expression](#) operands).

```
#include <expression.h>
```

Inheritance diagram for LiteralExpression:



Collaboration diagram for LiteralExpression:



Public Member Functions

- **LiteralExpression** (std::unique_ptr< [Literal](#) > &&l)
- bool **isComplete** () const override
- std::stringstream **getTextFormatted** () const override
- [Expression::Type](#) **getType** () const override
- void **accept** ([Visitor](#) *v) override
- const std::unique_ptr< [Literal](#) > & **getLiteral** () const

Additional Inherited Members

5.67.1 Detailed Description

Represents [Literal](#) in expression. Possible terminal node of expression tree (it does not have any other [Expression](#) operands).

5.67.2 Member Function Documentation

5.67.2.1 accept()

```
void LiteralExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.67.2.2 getTextFormatted()

```
std::stringstream LiteralExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.67.2.3 getType()

```
Expression::Type LiteralExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.67.2.4 isComplete()

```
bool LiteralExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

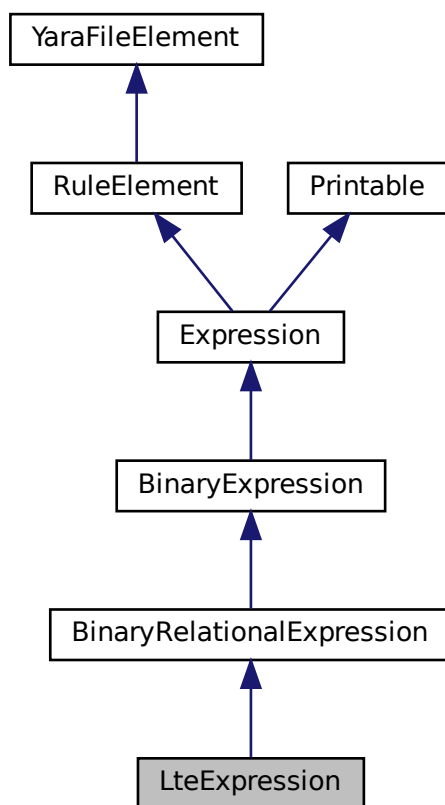
Implements [Expression](#).

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

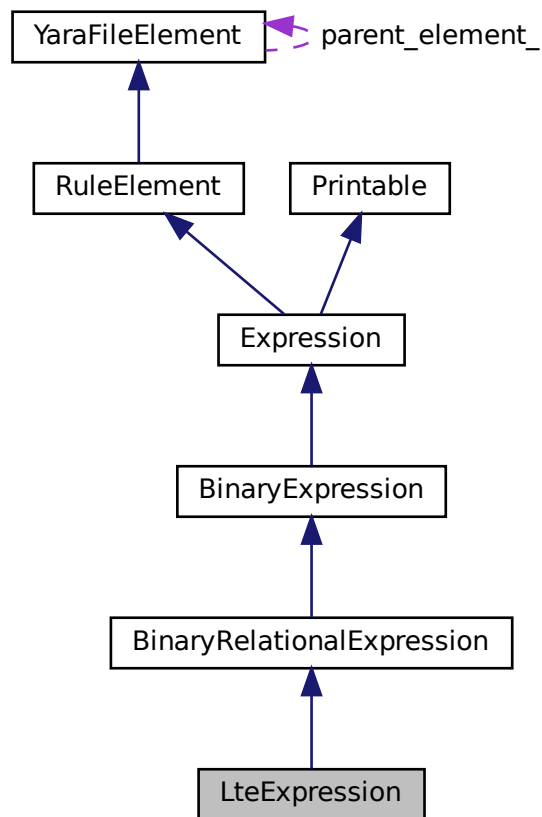
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.68 LteExpression Class Reference

Inheritance diagram for LteExpression:



Collaboration diagram for LteExpression:



Public Member Functions

- **LteExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string **opsign** () const override
- void **accept** (Visitor *v) override

Additional Inherited Members

5.68.1 Member Function Documentation

5.68.1.1 accept()

```
void LteExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.68.1.2 opsign()

```
std::string LtExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

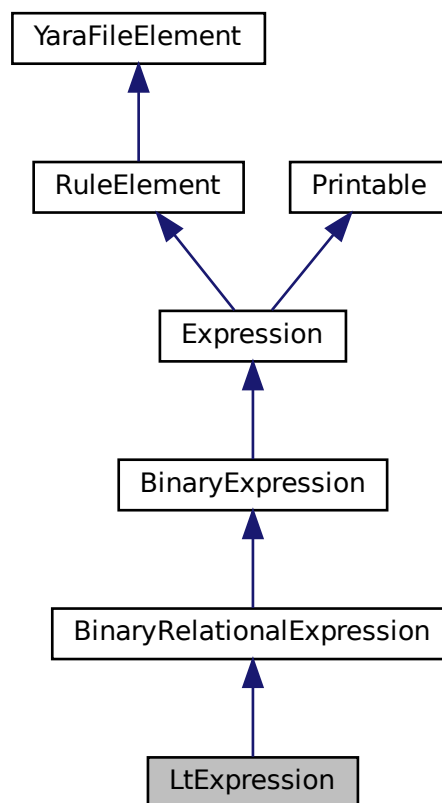
Implements [BinaryExpression](#).

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

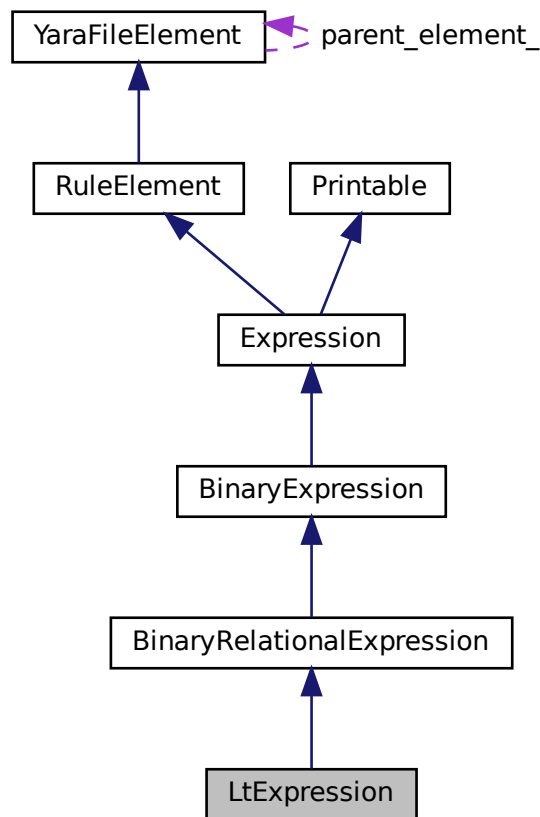
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.69 LtExpression Class Reference

Inheritance diagram for LtExpression:



Collaboration diagram for LtExpression:



Public Member Functions

- **LtExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.69.1 Member Function Documentation

5.69.1.1 [accept\(\)](#)

```
void LtExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.69.1.2 opsign()

```
std::string LtExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

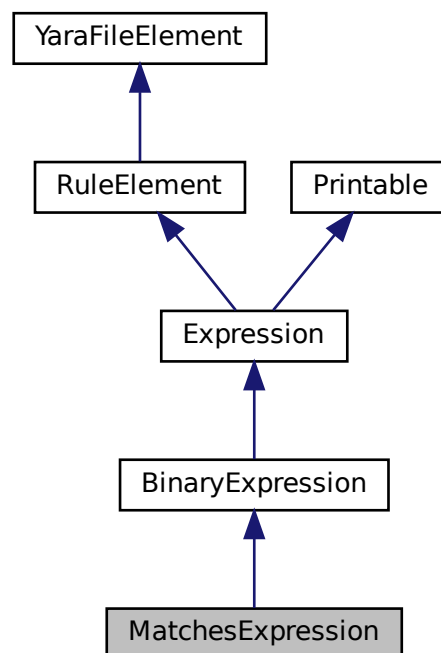
Implements [BinaryExpression](#).

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

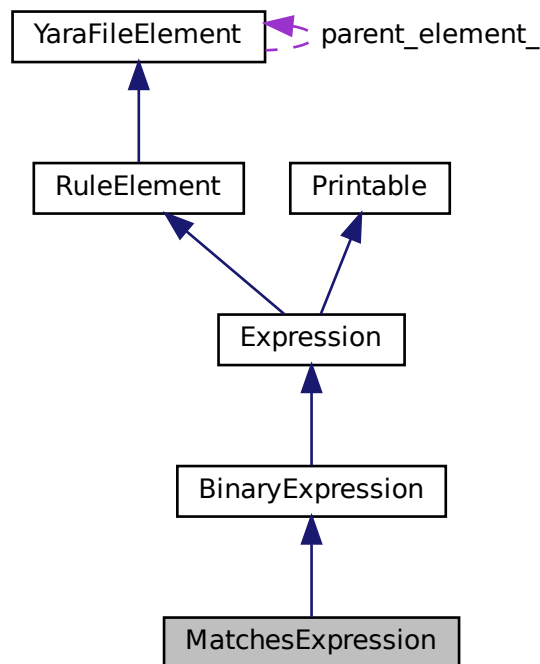
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.70 MatchesExpression Class Reference

Inheritance diagram for MatchesExpression:



Collaboration diagram for MatchesExpression:



Public Member Functions

- **MatchesExpression** (ExpressionPtr lop, ExpressionPtr rop)
- bool `isValid` (std::string_view &msg) const override
- `Expression::Type` `getType` () const override
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.70.1 Member Function Documentation

5.70.1.1 `accept()`

```
void MatchesExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.70.1.2 getType()

```
Expression::Type MatchesExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.70.1.3 isValid()

```
bool MatchesExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

5.70.1.4 opsign()

```
std::string MatchesExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

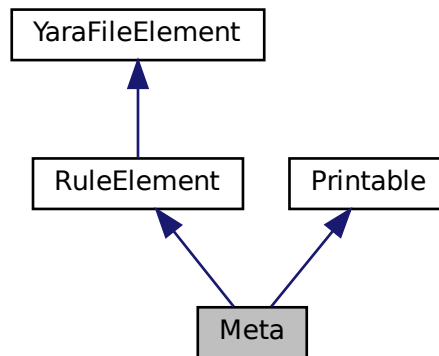
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.71 Meta Class Reference

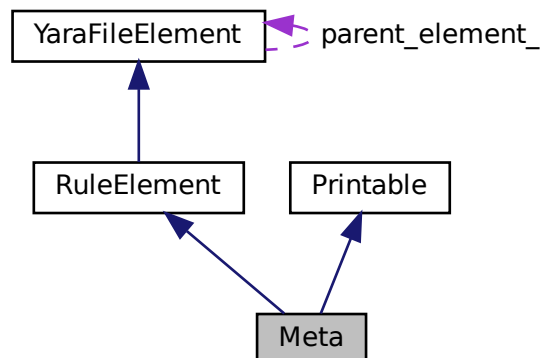
Represents meta information of rule.

```
#include <meta.h>
```

Inheritance diagram for Meta:



Collaboration diagram for Meta:



Public Member Functions

- **Meta** (const std::string &id, const std::string &value)
- **Meta** (const std::string &id, int64_t value)
- **Meta** (const std::string &id, bool value)
- const std::string & [getId](#) () const
- void [setId](#) (std::string &&id)
- const std::unique_ptr< [Literal](#) > & [getValue](#) () const
- void [setValue](#) (std::unique_ptr< [Literal](#) > &&literal)
- std::stringstream [getTextFormatted](#) () const

Additional Inherited Members

5.71.1 Detailed Description

Represents meta information of rule.

5.71.2 Member Function Documentation

5.71.2.1 getId()

```
const std::string & Meta::getId ( ) const
```

Returns id of the meta

Returns

Reference to string with meta id

5.71.2.2 getTextFormatted()

```
std::stringstream Meta::getTextFormatted ( ) const [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.71.2.3 getValue()

```
const std::unique_ptr< Literal > & Meta::getValue ( ) const
```

Returns literal with meta value

5.71.2.4 setId()

```
void Meta::setId (
    std::string && id )
```

Sets the id of the meta

5.71.2.5 setValue()

```
void Meta::setValue (
    std::unique_ptr< Literal > && literal )
```

Sets the literal, that represents the value of meta

Note

For literal in YARA are valid only bool, string and int literal, if other type is given, [YaramodErrorException](#) is thrown

< Undefined literal is also allowed to be more tolerant against against

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

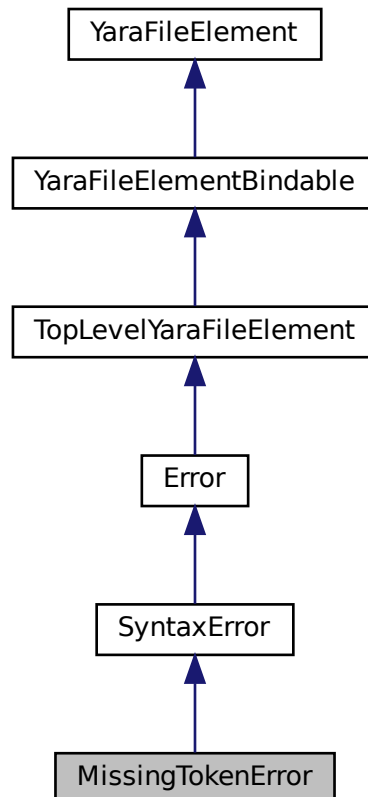
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[meta.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[meta.cpp](#)

5.72 MissingTokenError Class Reference

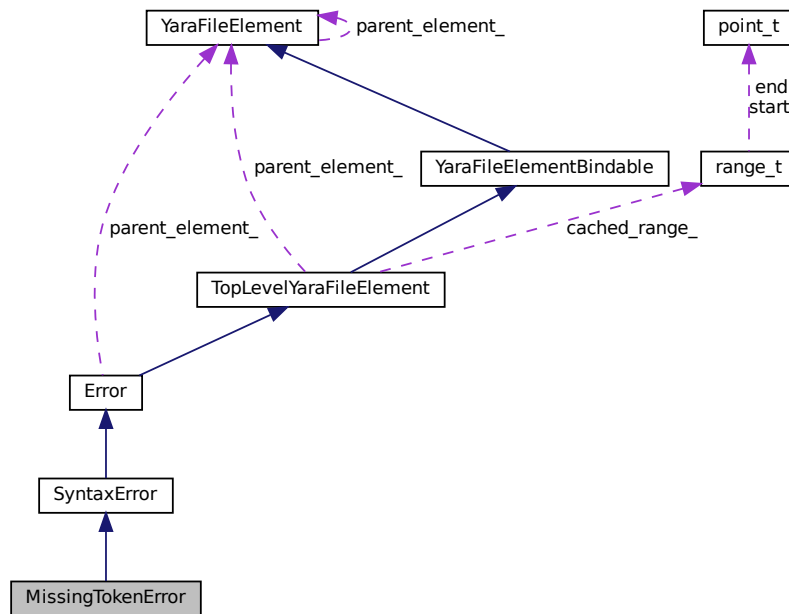
Class for syntax error caused missing token.

```
#include <error.h>
```

Inheritance diagram for MissingTokenError:



Collaboration diagram for MissingTokenError:



Public Member Functions

- **MissingTokenError** (const std::string &missing_str, const offset_t &offset, const size_t &len)
- [MissingTokenErrorException](#) exception ()

Additional Inherited Members

5.72.1 Detailed Description

Class for syntax error caused missing token.

Note

These errors are identified by the tree-sitter parser when it is clear what is missing

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

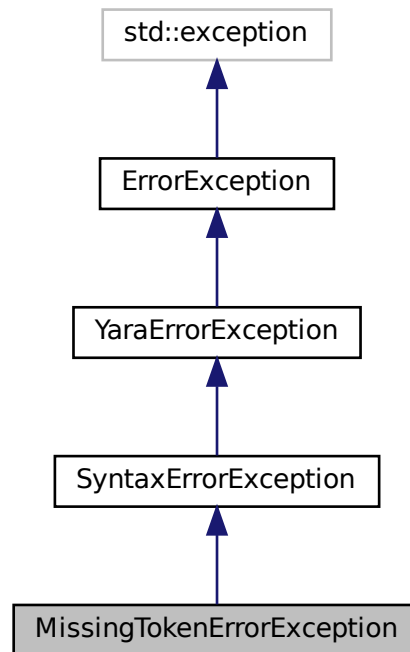
- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h`

5.73 MissingTokenErrorException Class Reference

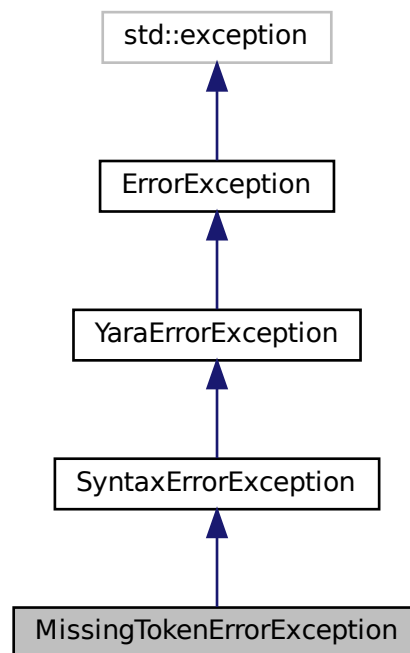
Exception representing missing token in parsed source code.

```
#include <error.h>
```

Inheritance diagram for MissingTokenErrorException:



Collaboration diagram for MissingTokenErrorException:



Public Member Functions

- **MissingTokenErrorException** (offset_t offset, size_t len, std::string msg={})

5.73.1 Detailed Description

Exception representing missing token in parsed source code.

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.74 Module Class Reference

Class representing module (its description)

```
#include <module.h>
```

Public Member Functions

- **Module** (const std::string &path)
- **Module** (std::string_view path)
- const std::string & **getName** () const
- void **load** ()
- const std::unique_ptr< json > & **getData** () const
- void **merge** (const std::shared_ptr< **Module** > &other)

Static Public Member Functions

- static void **mergeJson** (json *dst, const json *src)
- static **Expression::Type jsonToExprType** (const json *json_type)
- static const json * **findInArray** (const json *json, std::string_view obj_name)
- static const json * **findField** (const json *json, std::string_view field_name)
- static **Expression::Type getDataType** (const json *ctx_obj)
- static const json * **qualifyObject** (std::string_view qual_str, const json *obj)
- static **Expression::Type getElementDataType** (const json *ctx_obj)
- static **Symbol::Type getElementType** (const json *ctx_obj)
- static void **buildModuleContext** (const std::shared_ptr< **Symbol** > &identifier, const json *ctx_obj)

5.74.1 Detailed Description

Class representing module (its description)

5.74.2 Member Function Documentation

5.74.2.1 buildModuleContext()

```
void Module::buildModuleContext (
    const std::shared_ptr< Symbol > & identifier,
    const json * ctx_obj ) [static]
```

Builds semantic context of given **Symbol** object from given JSON object < Get top level id (the name of module)

< Try to find corresponding module

< Qualify reference type

5.74.2.2 findField()

```
const json * Module::findField (
    const json * json,
    std::string_view field_name ) [static]
```

Finds field with given name in JSON object

Returns

pointer to field value with given name or nullptr

5.74.2.3 findInArray()

```
const json * Module::findInArray (
    const json * json,
    std::string_view obj_name ) [static]
```

Finds object with given name in JSON array

Returns

pointer to object with given name or nullptr

5.74.2.4 getData()

```
const std::unique_ptr< json > & Module::getData ( ) const
```

Returns pointer to module data (JSON object)

5.74.2.5 getDataType()

```
Expression::Type Module::getDataType (
    const json * ctx_obj ) [static]
```

Gets data type of given JSON object

Returns

Data type represented as [Expression::Type](#)

< Data type is stored in object in 'structure' field

5.74.2.6 getElementDataType()

```
Expression::Type Module::getElementDataType (
    const json * ctx_obj ) [static]
```

Find data type of element of array in given JSON object

Note

Wrapper above [Module::getDataType](#)

5.74.2.7 getElementType()

```
Symbol::Type Module::getElementType (
    const json * ctx_obj ) [static]
```

Finds type of element of array in given JSON object < If element is reference kind, the type must be qualified

< Get top level id (module name)

< Try to find module with this top level id

< Do qualification

5.74.2.8 jsonToExprType()

```
Expression::Type Module::jsonToExprType (
    const json * json_type ) [static]
```

Converts string in JSON to [Expression::Type](#)

5.74.2.9 load()

```
void Module::load ( )
```

Parses file in JSON format with module description to JSON object to be easily traversed

5.74.2.10 merge()

```
void Module::merge (
    const std::shared_ptr< Module > & other )
```

Merges two modules (their data)

5.74.2.11 mergeJson()

```
void Module::mergeJson (
    json * dst,
    const json * src ) [static]
```

Recursively merges two JSON objects < If item is in the both files merge it recursively

< If it is not in dst object add it

< JSON object is array type

< Only objects are processed (there are not directly nested arrays in modules)

< Ty to find corresponding object in dst

5.74.2.12 qualifyObject()

```
const json * Module::qualifyObject (
    std::string_view qual_str,
    const json * obj ) [static]
```

Tries to find JSON object that is specified by given string

Returns

JSON object with context that corresponds to the given string

< Get top level name

< Try to find top level name in attributes

< Top level is also the last id in qual_str

< Update qualifier string

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

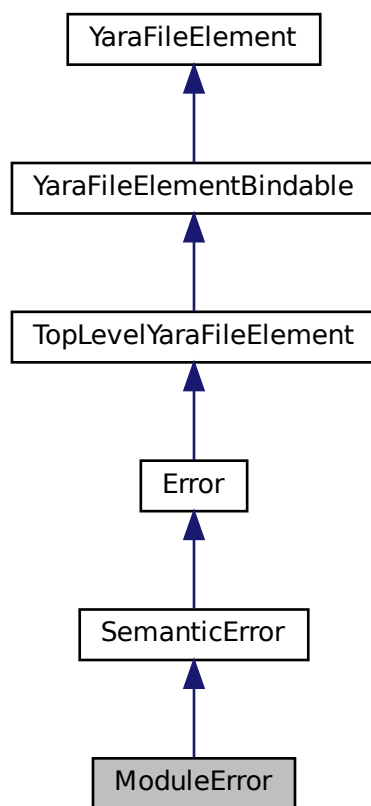
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[module.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[module.cpp](#)

5.75 ModuleError Class Reference

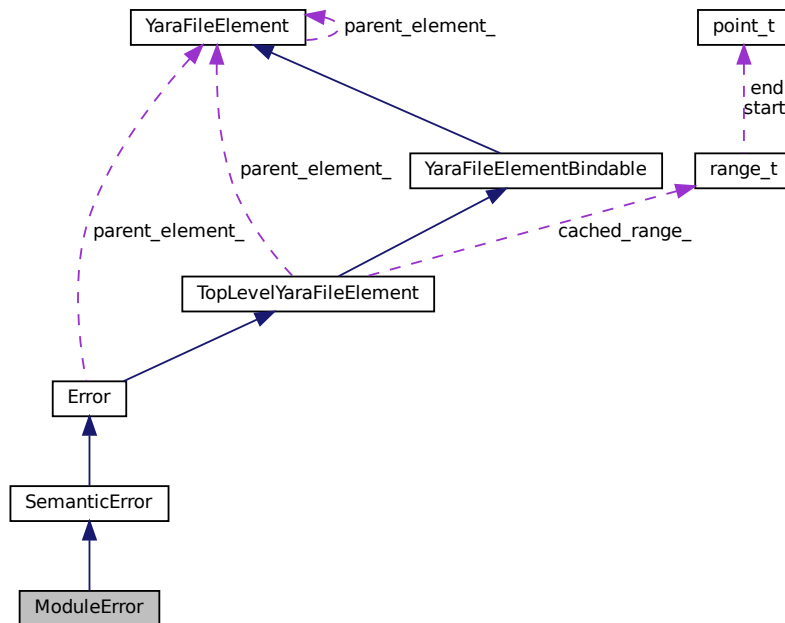
[Error](#) that occurred in module import.

```
#include <error.h>
```

Inheritance diagram for ModuleError:



Collaboration diagram for ModuleError:



Public Member Functions

- **ModuleError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [ModuleErrorException](#) **exception** ()

Additional Inherited Members

5.75.1 Detailed Description

[Error](#) that occurred in module import.

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

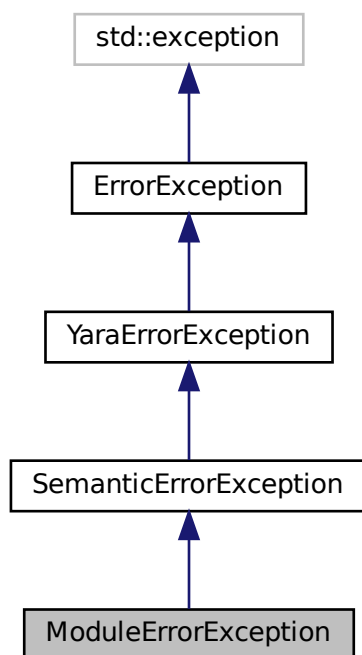
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.76 ModuleErrorException Class Reference

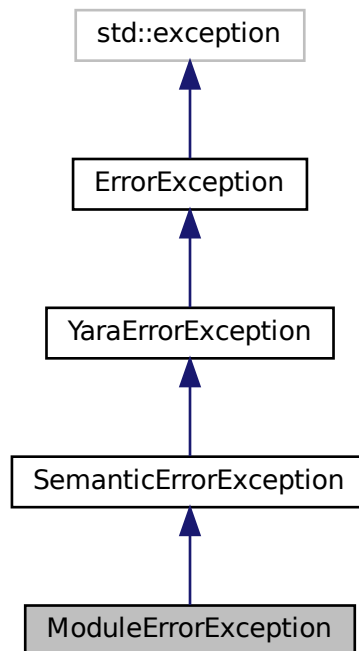
Exception for errors, that occurred in module import

```
#include <error.h>
```

Inheritance diagram for ModuleErrorException:



Collaboration diagram for ModuleErrorException:



Public Member Functions

- **ModuleErrorException** (offset_t offset, size_t len, std::string msg={})

5.76.1 Detailed Description

Exception for errors, that occurred in module import

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.77 ModuleProvider Class Reference

Singleton class that provides access to loaded modules. Modules are loaded, when [Yaramod](#) class is instantiated.

```
#include <module.h>
```

Public Types

- using **ModuleMap** = std::unordered_map< std::string, std::shared_ptr< [Module](#) >, [string_hash_t](#), std::equal_to<> >
- using **SymbolCache** = std::unordered_map< std::string, const json *, [string_hash_t](#), std::equal_to<> >

Public Member Functions

- bool [hasModule](#) (const std::string &name) const
- bool **hasModule** (const std::string_view name) const
- void [addModule](#) (const std::shared_ptr< [Module](#) > &module)
- std::shared_ptr< [Module](#) > [getModule](#) (const std::string &name) const
- std::shared_ptr< [Module](#) > **getModule** (std::string_view name) const
- void [clearCache](#) ()
- const json * [findCached](#) (std::string_view full_symbol_name) const
- void [cacheSymbol](#) (const std::string &parent_namespace, const json *symbol_ctx)

Static Public Member Functions

- static [ModuleProvider](#) & [getInstance](#) ()

5.77.1 Detailed Description

Singleton class that provides access to loaded modules. Modules are loaded, when [Yaramod](#) class is instantiated.

5.77.2 Member Function Documentation

5.77.2.1 addModule()

```
void ModuleProvider::addModule (
    const std::shared_ptr< Module > & module )
```

Adds module to [ModuleProvider](#) (this module can be accessed later by [getModule](#))

5.77.2.2 cacheSymbol()

```
void ModuleProvider::cacheSymbol (
    const std::string & parent_namespace,
    const json * symbol_ctx )
```

Creates entry in cache for the given symbol json ctx

5.77.2.3 clearCache()

```
void ModuleProvider::clearCache ( )
```

Clears the symbol cache

5.77.2.4 findCached()

```
const json * ModuleProvider::findCached (
    std::string_view full_symbol_name ) const
```

Find cached symbol context (by full symbol name)

5.77.2.5 getInstance()

```
ModuleProvider & ModuleProvider::getInstance ( ) [static]
```

Returns singleton instance of [ModuleProvider](#) < Singleton instance (it is destroyed automatically)

5.77.2.6 getModule()

```
std::shared_ptr< Module > ModuleProvider::getModule (
    const std::string & name ) const
```

Returns pointer to module with given name, that was stored to module provider. [Module](#), that was returned by this function must not be deleted! ([ModuleProvider](#) deletes it automatically)

5.77.2.7 hasModule()

```
bool ModuleProvider::hasModule (
    const std::string & name ) const
```

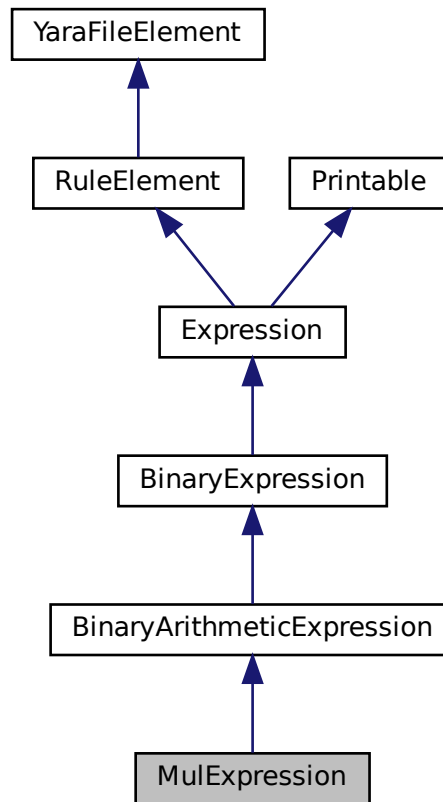
Checks whether module with given name was loaded to module provider or not

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

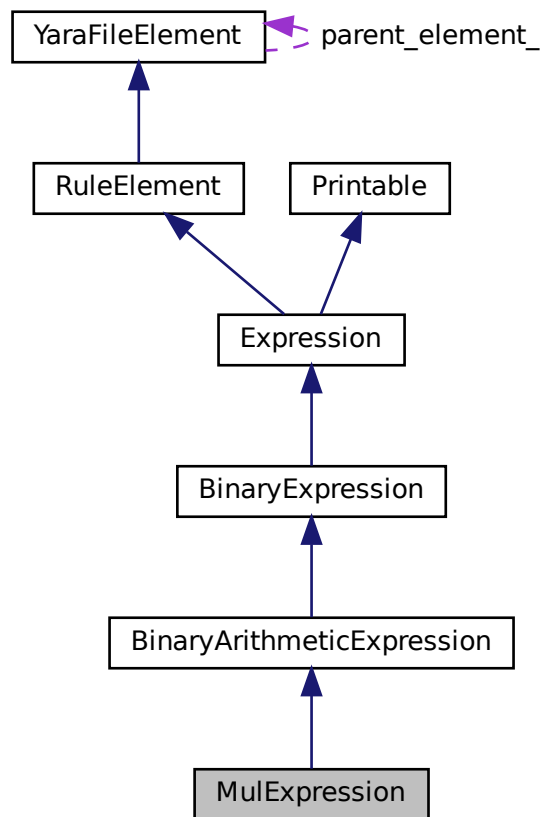
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/module.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/module.cpp](#)

5.78 MulExpression Class Reference

Inheritance diagram for MulExpression:



Collaboration diagram for MulExpression:



Public Member Functions

- **MulExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.78.1 Member Function Documentation

5.78.1.1 `accept()`

```
void MulExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.78.1.2 opsign()

```
std::string MulExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

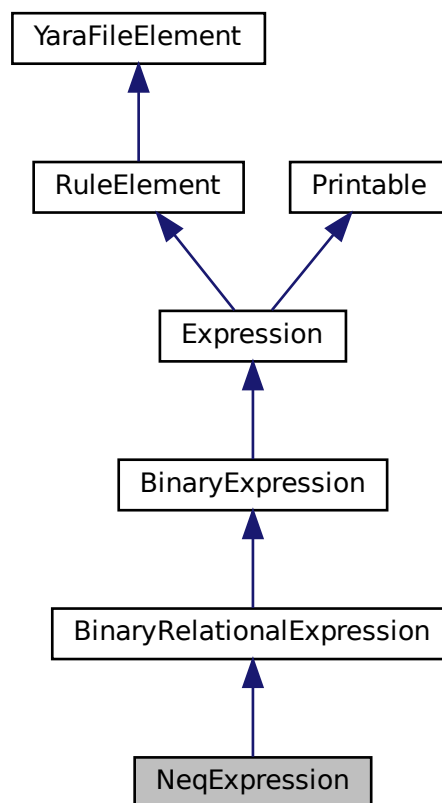
Implements [BinaryExpression](#).

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

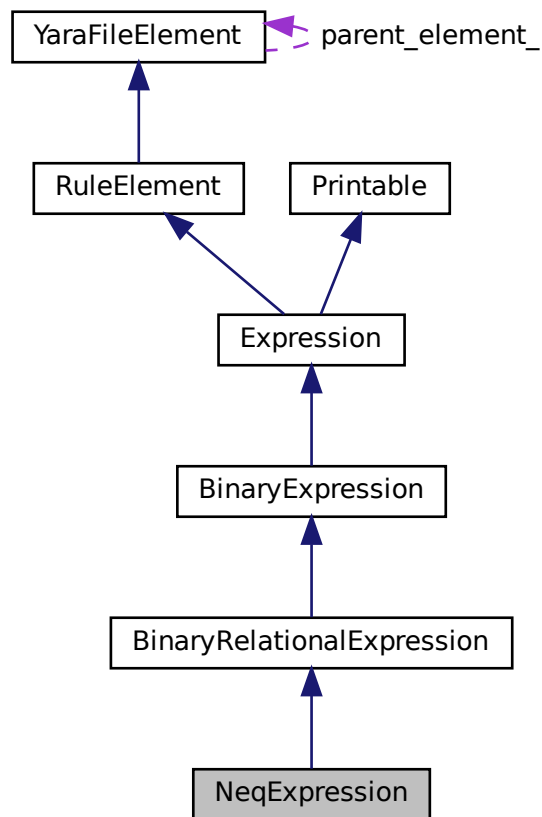
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.79 NeqExpression Class Reference

Inheritance diagram for NeqExpression:



Collaboration diagram for NeqExpression:



Public Member Functions

- **NeqExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.79.1 Member Function Documentation

5.79.1.1 `accept()`

```
void NeqExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.79.1.2 opsign()

```
std::string NeqExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.80 node_w_depth_t Struct Reference

Public Attributes

- `size_t` **depth**
- `TSNode` **node**

5.80.1 Detailed Description

Auxiliary structure for associating node with its depth (used in methods for traversing)

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

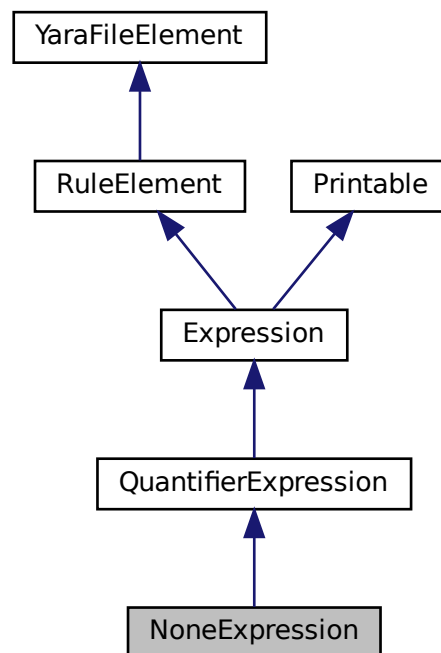
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter.cpp](#)

5.81 NoneExpression Class Reference

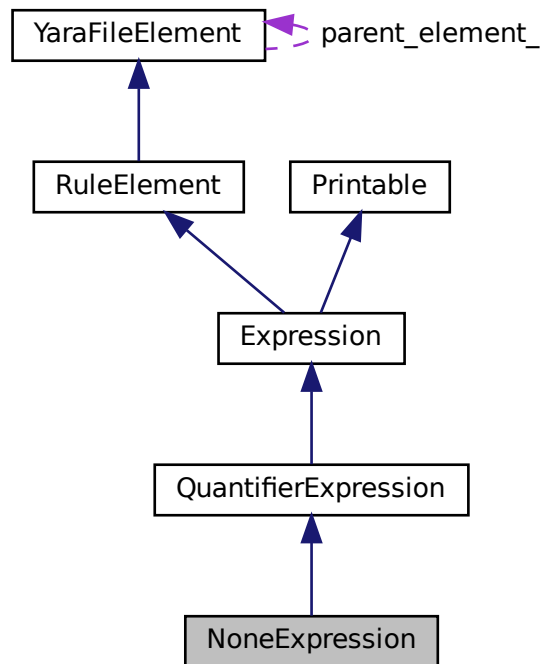
Represents 'none' keyword.

```
#include <expression.h>
```

Inheritance diagram for NoneExpression:



Collaboration diagram for NoneExpression:



Public Member Functions

- `std::stringstream` [getTextFormatted](#) () const override
- `void` [accept](#) ([Visitor](#) *v) override

Additional Inherited Members

5.81.1 Detailed Description

Represents 'none' keyword.

5.81.2 Member Function Documentation

5.81.2.1 `accept()`

```
void NoneExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.81.2.2 getTextFormatted()

```
std::stringstream NoneExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

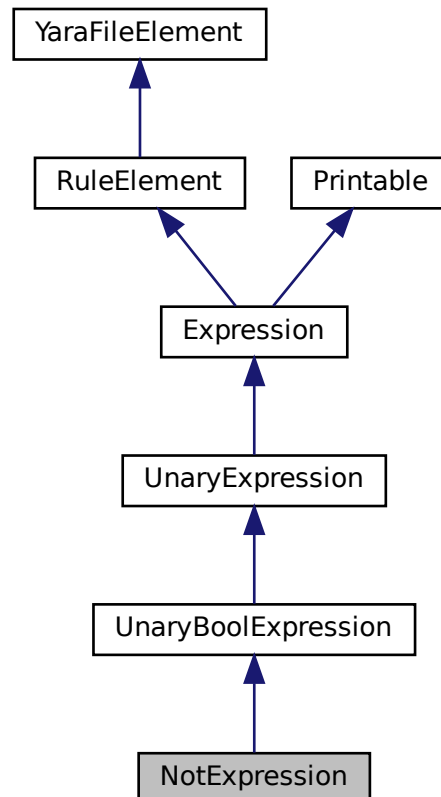
Implements [Printable](#).

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

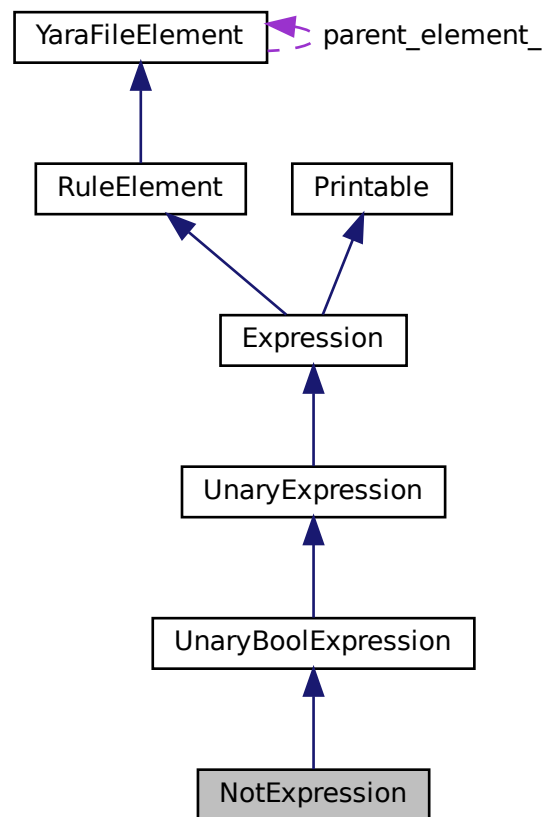
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.82 NotExpression Class Reference

Inheritance diagram for NotExpression:



Collaboration diagram for NotExpression:



Public Member Functions

- **NotExpression** (ExpressionPtr op)
- std::string **opsign** () const override
- void **accept** (Visitor *v) override

Additional Inherited Members

5.82.1 Member Function Documentation

5.82.1.1 accept()

```
void NotExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

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

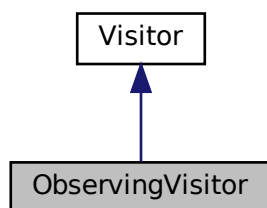
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.83 ObservingVisitor Class Reference

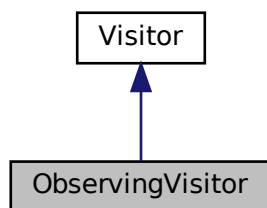
Concrete [Visitor](#) class for observing expression tree.

```
#include <observing_visitor.h>
```

Inheritance diagram for ObservingVisitor:



Collaboration diagram for ObservingVisitor:



Public Member Functions

- void [visitBinaryOperatorDefault](#) ([BinaryExpression](#) *e)
- void [visitUnaryOperatorDefault](#) ([UnaryExpression](#) *e)
- void [visitSetDefault](#) ([SetExpression](#) *e)
- void [start](#) ([Expression](#) *e)
- virtual void [visitRange](#) ([Range](#) *r) override
- virtual void [visitEnum](#) ([Enum](#) *e) override
- virtual void [visitParentheses](#) ([ParenthesesExpression](#) *e) override
- virtual void [visitIn](#) ([InExpression](#) *e) override
- virtual void [visitMatches](#) ([MatchesExpression](#) *e) override
- virtual void [visitAt](#) ([AtExpression](#) *e) override
- virtual void [visitAdd](#) ([AddExpression](#) *e) override
- virtual void [visitSub](#) ([SubExpression](#) *e) override
- virtual void [visitMul](#) ([MulExpression](#) *e) override
- virtual void [visitRemainder](#) ([RemainderExpression](#) *e) override
- virtual void [visitDiv](#) ([DivExpression](#) *e) override
- virtual void [visitLeftShift](#) ([LeftShiftExpression](#) *e) override
- virtual void [visitRightShift](#) ([RightShiftExpression](#) *e) override
- virtual void [visitBitwiseOr](#) ([BitwiseOrExpression](#) *e) override
- virtual void [visitBitwiseAnd](#) ([BitwiseAndExpression](#) *e) override
- virtual void [visitBitwiseXor](#) ([BitwiseXorExpression](#) *e) override
- virtual void [visitLt](#) ([LtExpression](#) *e) override
- virtual void [visitLte](#) ([LteExpression](#) *e) override
- virtual void [visitGt](#) ([GtExpression](#) *e) override
- virtual void [visitGte](#) ([GteExpression](#) *e) override
- virtual void [visitEq](#) ([EqExpression](#) *e) override
- virtual void [visitNeq](#) ([NeqExpression](#) *e) override
- virtual void [visitContains](#) ([ContainsExpression](#) *e) override
- virtual void [visitIContains](#) ([IContainsExpression](#) *e) override
- virtual void [visitStartsWith](#) ([StartsWithExpression](#) *e) override
- virtual void [visitIStartsWith](#) ([IStartsWithExpression](#) *e) override
- virtual void [visitEndsWith](#) ([EndsWithExpression](#) *e) override
- virtual void [visitIEndsWith](#) ([IEndsWithExpression](#) *e) override
- virtual void [visitIEq](#) ([IEqExpression](#) *e) override
- virtual void [visitAnd](#) ([AndExpression](#) *e) override
- virtual void [visitOr](#) ([OrExpression](#) *e) override
- virtual void [visitDefined](#) ([DefinedExpression](#) *e) override
- virtual void [visitUnaryMinus](#) ([UnaryMinusExpression](#) *e) override
- virtual void [visitBitwiseNot](#) ([BitwiseNotExpression](#) *e) override
- virtual void [visitNot](#) ([NotExpression](#) *e) override
- virtual void [visitLiteral](#) ([LiteralExpression](#) *e) override
- virtual void [visitSize](#) ([SizeExpression](#) *e) override
- virtual void [visitRegex](#) ([RegexExpression](#) *e) override
- virtual void [visitString](#) ([StringExpression](#) *e) override
- virtual void [visitStringWildcard](#) ([StringWildcardExpression](#) *e) override
- virtual void [visitStringCount](#) ([StringCountExpression](#) *e) override
- virtual void [visitStringOffset](#) ([StringOffsetExpression](#) *e) override
- virtual void [visitStringMatchLength](#) ([StringMatchLengthExpression](#) *e) override
- virtual void [visitPlainSymbol](#) ([PlainSymbol](#) *symbol) override
- virtual void [visitArray](#) ([ArrayExpression](#) *e) override
- virtual void [visitStruct](#) ([StructExpression](#) *e) override
- virtual void [visitFunctionCall](#) ([FunctionCallExpression](#) *e) override
- virtual void [visitVariable](#) ([VariableExpression](#) *e) override
- virtual void [visitRuleWildcard](#) ([RuleWildcardExpression](#) *e) override

- virtual void **visitNone** ([NoneExpression](#) *e) override
- virtual void **visitAny** ([AnyExpression](#) *e) override
- virtual void **visitAll** ([AllExpression](#) *e) override
- virtual void **visitPercentQuantifier** ([PercentQuantifierExpression](#) *e) override
- virtual void **visitThem** ([ThemExpression](#) *e) override
- virtual void **visitStringSet** ([StringSetExpression](#) *e) override
- virtual void **visitRuleSet** ([RuleSetExpression](#) *e) override
- virtual void **visitExpressionSet** ([ExpressionSetExpression](#) *e) override
- virtual void **visitOf** ([OfExpression](#) *e) override
- virtual void **visitFor** ([ForExpression](#) *e) override
- virtual void **visitForInt** ([ForIntExpression](#) *e) override
- virtual void **visitVarList** ([VarListExpression](#) *e) override

5.83.1 Detailed Description

Concrete [Visitor](#) class for observing expression tree.

Tree is stored in condition member of rule or in value member of internal variables - Avast specific). It is base class for user-defined observing visitors

5.83.2 Member Function Documentation

5.83.2.1 start()

```
void ObservingVisitor::start (
    Expression * e )
```

Executes expression traversal above any [Expression](#) subclass

5.83.2.2 visitBinaryOperatorDefault()

```
void ObservingVisitor::visitBinaryOperatorDefault (
    BinaryExpression * e )
```

Helper function to reduce redundant code of visit<BinaryExpression> methods

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

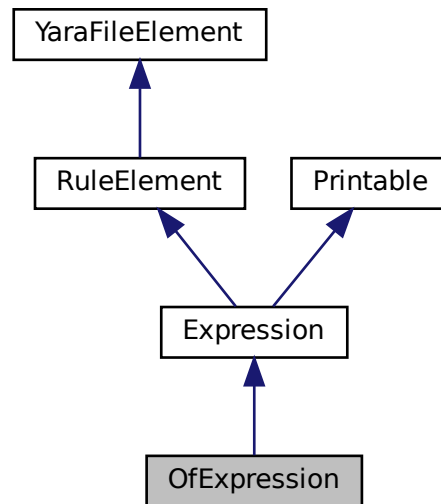
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[observing_visitor.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/visitor/[observing_visitor.cpp](#)

5.84 OfExpression Class Reference

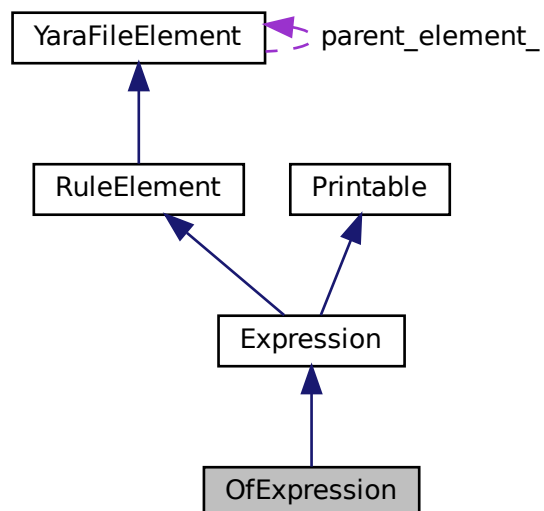
Represents all variants of 'of' operator in YARA.

```
#include <expression.h>
```

Inheritance diagram for OfExpression:



Collaboration diagram for OfExpression:



Public Member Functions

- **OfExpression** (ExpressionPtr quantifier, ExpressionPtr set, std::shared_ptr< [Range](#) > range)
- bool [isValid](#) (std::string_view &) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- [Expression::Type](#) [getType](#) () const override
- void [accept](#) (Visitor *v) override
- const ExpressionPtr & [getQuantifier](#) () const
- const ExpressionPtr & [getSet](#) () const

Additional Inherited Members

5.84.1 Detailed Description

Represents all variants of 'of' operator in YARA.

5.84.2 Member Function Documentation

5.84.2.1 [accept\(\)](#)

```
void OfExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.84.2.2 [areOperandsValid\(\)](#)

```
bool OfExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.84.2.3 [getQuantifier\(\)](#)

```
const ExpressionPtr & OfExpression::getQuantifier ( ) const
```

Returns quantifier of 'of' operator

5.84.2.4 `getSet()`

```
const ExpressionPtr & OfExpression::getSet ( ) const
```

Returns set of 'of' operator

5.84.2.5 `getTextFormatted()`

```
std::stringstream OfExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.84.2.6 `getType()`

```
Expression::Type OfExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.84.2.7 `isComplete()`

```
bool OfExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.84.2.8 `isValid()`

```
bool OfExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<code>msg</code>	error message, that describes an error
------------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h`
- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp`

5.85 offset_edit_range_t Struct Reference

Structure for lossless storing data about atomic edits in unified form.

```
#include <types.h>
```

Public Types

- enum class [EditType](#) { [MOD](#) , [DEL](#) , [INS](#) }
- Type of edit.*

Public Member Functions

- [offset_edit_range_t](#) (const [offset_range_t](#) &offset_range, [EditType](#) edit_type=[EditType::MOD](#))
- [offset_edit_range_t](#) (const [offset_t](#) &start_offset, const [offset_t](#) &end_offset, [EditType](#) edit_type=[EditType::MOD](#))
- [size_t](#) [len](#) ()
- bool [operator<](#) (const [offset_edit_range_t](#) &right_range)

Public Attributes

- [offset_t](#) [start](#)
- [offset_t](#) [end](#)
- [EditType](#) [type](#) = [EditType::MOD](#)

5.85.1 Detailed Description

Structure for lossless storing data about atomic edits in unified form.

5.85.2 Member Enumeration Documentation

5.85.2.1 EditType

```
enum offset\_edit\_range\_t::EditType [strong]
```

Type of edit.

Enumerator

MOD	Unspecified modification.
DEL	Deletion.
INS	Insertion.

5.85.3 Member Function Documentation

5.85.3.1 len()

```
size_t offset_edit_range_t::len ( ) [inline]
```

Computes the length of range

Returns

the length of the range

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[types.h](#)

5.86 offset_range_t Struct Reference

Offset range for specifying the part of file, that was for example modified, deleted etc.

```
#include <types.h>
```

Public Member Functions

- bool **operator**< (const [offset_range_t](#) &right_range)

Public Attributes

- offset_t **start**
- offset_t **end**

5.86.1 Detailed Description

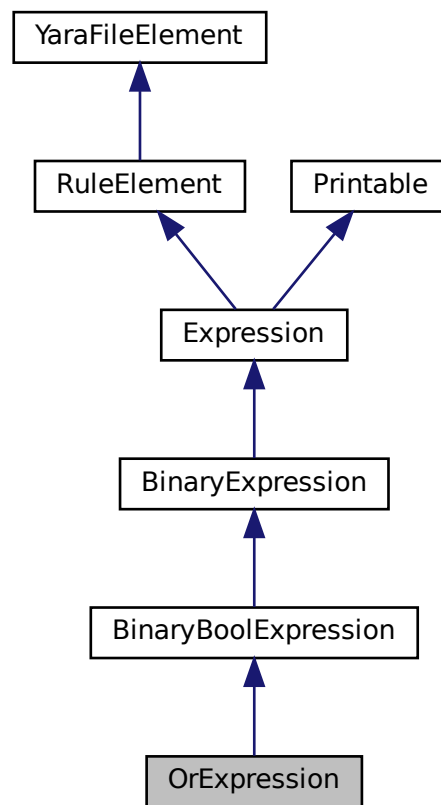
Offset range for specifying the part of file, that was for example modified, deleted etc.

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

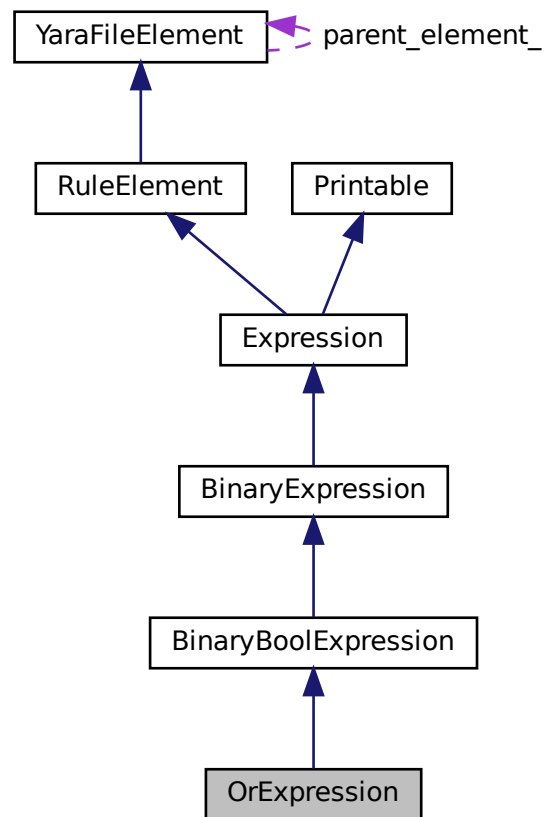
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[types.h](#)

5.87 OrExpression Class Reference

Inheritance diagram for OrExpression:



Collaboration diagram for OrExpression:



Public Member Functions

- **OrExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.87.1 Member Function Documentation

5.87.1.1 [accept\(\)](#)

```
void OrExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.87.1.2 opsign()

```
std::string OrExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

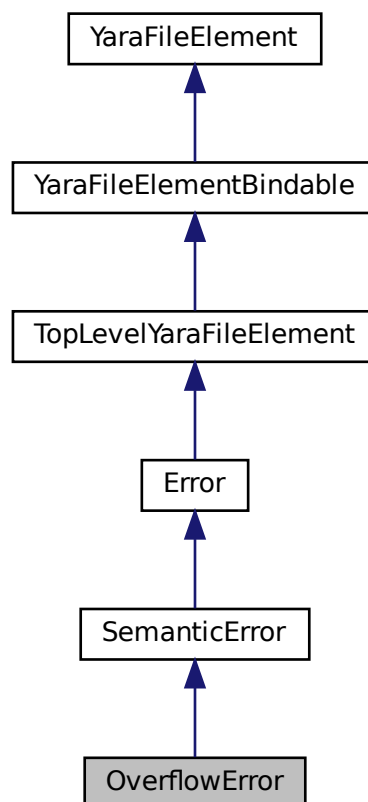
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.88 OverflowError Class Reference

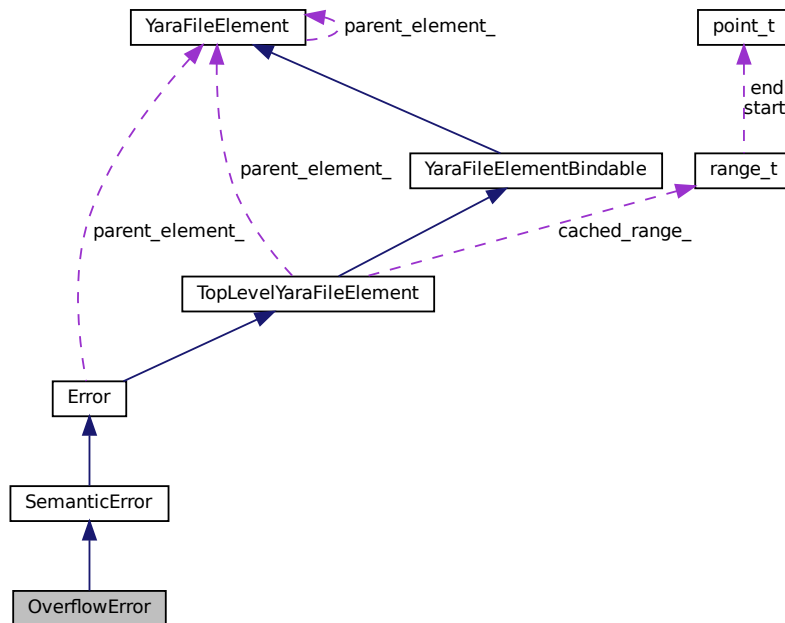
Overflow error (e. g. integer overflow)

```
#include <error.h>
```

Inheritance diagram for OverflowError:



Collaboration diagram for OverflowError:



Public Member Functions

- **OverflowError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [OverflowErrorException](#) exception ()

Additional Inherited Members

5.88.1 Detailed Description

Overflow error (e. g. integer overflow)

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

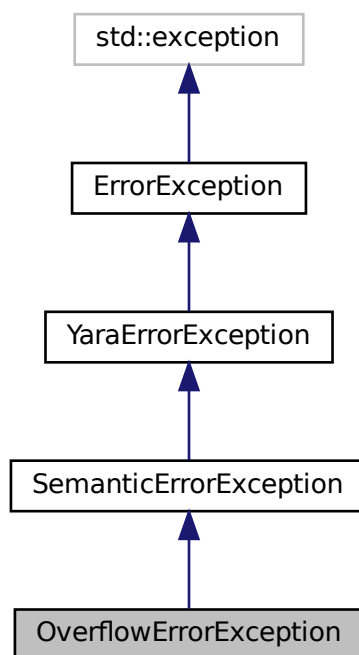
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.89 OverflowErrorException Class Reference

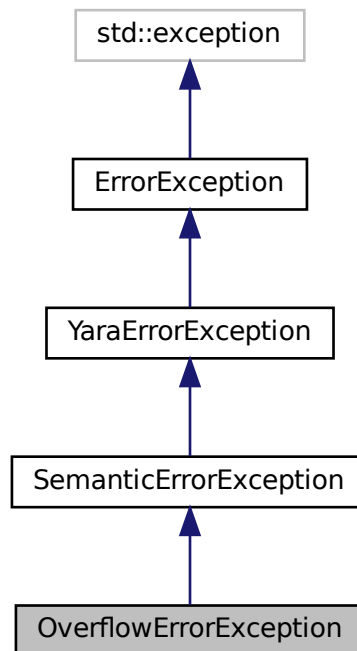
Exception for overflows.

```
#include <error.h>
```

Inheritance diagram for OverflowErrorException:



Collaboration diagram for OverflowErrorException:



Public Member Functions

- **OverflowErrorException** (offset_t offset, size_t len, std::string msg={})

5.89.1 Detailed Description

Exception for overflows.

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

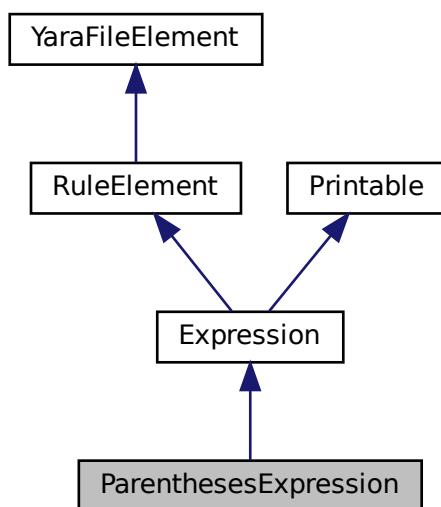
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.90 ParenthesesExpression Class Reference

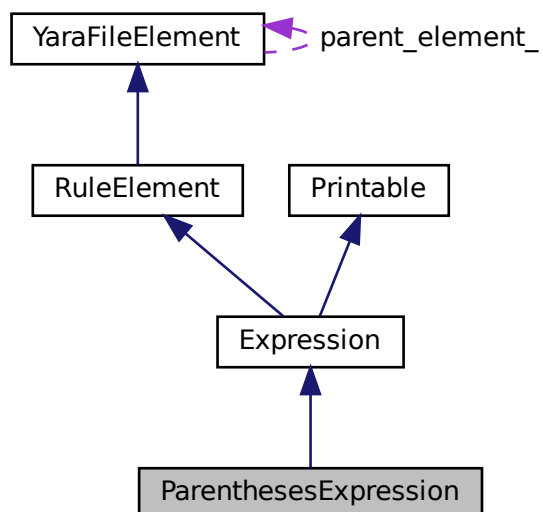
Base class for parentheses () in expressions.

```
#include <expression.h>
```


Inheritance diagram for ParenthesesExpression:



Collaboration diagram for ParenthesesExpression:



Public Member Functions

- **ParenthesesExpression** (ExpressionPtr inner_expr)

- bool [isValid](#) (std::string_view &msg) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- [Expression::Type](#) [getType](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) ([Visitor](#) *v) override
- const ExpressionPtr & [getInnerExpression](#) () const

Additional Inherited Members

5.90.1 Detailed Description

Base class for parentheses () in expressions.

5.90.2 Member Function Documentation

5.90.2.1 [accept\(\)](#)

```
void ParenthesesExpression::accept (  
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.90.2.2 [areOperandsValid\(\)](#)

```
bool ParenthesesExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.90.2.3 [getTextFormatted\(\)](#)

```
std::stringstream ParenthesesExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.90.2.4 getType()

```
Expression::Type ParenthesesExpression::getType ( ) const [override], [virtual]
```

Only propagates type of inner expression (or returns undefined type if it does not have any)

Implements [Expression](#).

5.90.2.5 isComplete()

```
bool ParenthesesExpression::isComplete ( ) const [override], [virtual]
```

Checks whether expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.90.2.6 isValid()

```
bool ParenthesesExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Performs check of validity of inner expression

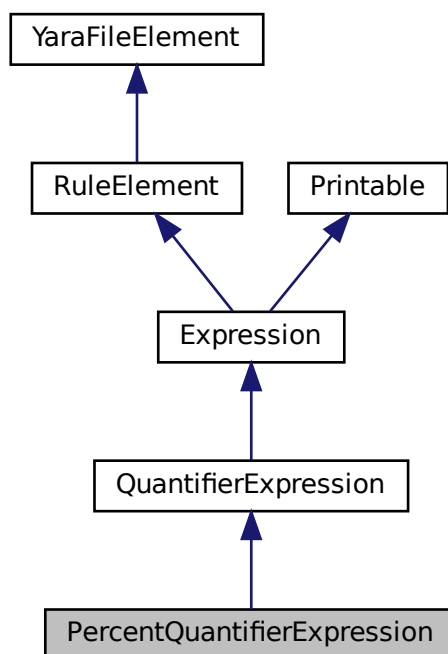
Implements [Expression](#).

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

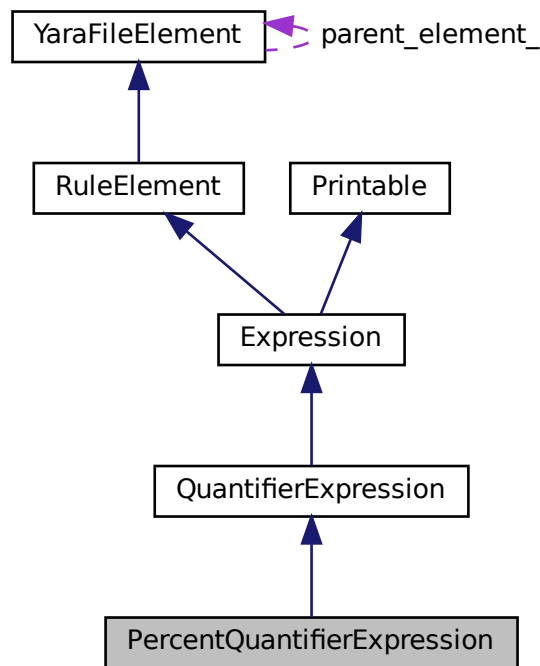
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.91 PercentQuantifierExpression Class Reference

Inheritance diagram for PercentQuantifierExpression:



Collaboration diagram for PercentQuantifierExpression:



Public Member Functions

- **PercentQuantifierExpression** (ExpressionPtr value)
- bool [isValid](#) (std::string_view &msg) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- [Expression::Type](#) [getType](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) (Visitor *v) override
- const ExpressionPtr & [getValue](#) () const

Additional Inherited Members

5.91.1 Member Function Documentation

5.91.1.1 accept()

```
void PercentQuantifierExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.91.1.2 areOperandsValid()

```
bool PercentQuantifierExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Reimplemented from [QuantifierExpression](#).

5.91.1.3 getTextFormatted()

```
std::stringstream PercentQuantifierExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.91.1.4 getType()

```
Expression::Type PercentQuantifierExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Reimplemented from [QuantifierExpression](#).

5.91.1.5 isComplete()

```
bool PercentQuantifierExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Reimplemented from [QuantifierExpression](#).

5.91.1.6 isValid()

```
bool PercentQuantifierExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

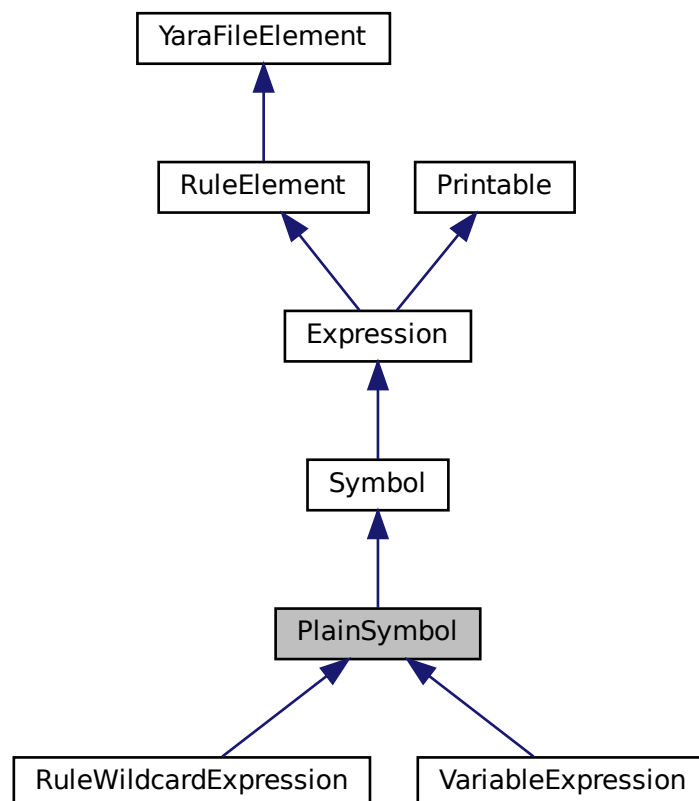
Reimplemented from [QuantifierExpression](#).

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

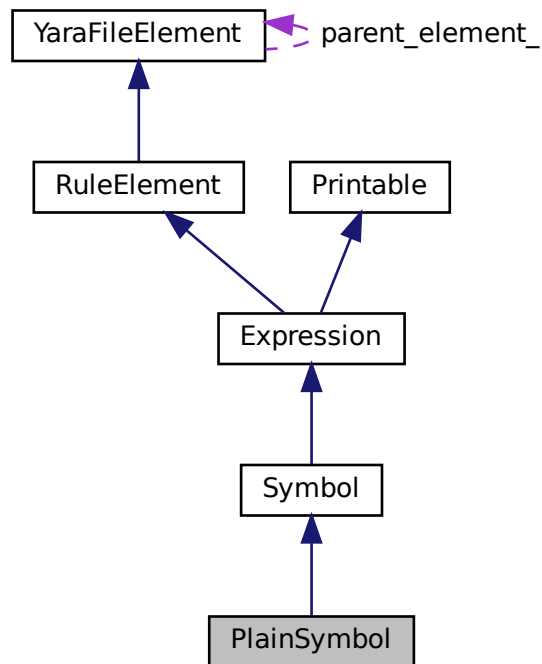
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.92 PlainSymbol Class Reference

Inheritance diagram for PlainSymbol:



Collaboration diagram for PlainSymbol:



Public Member Functions

- **PlainSymbol** (std::string_view id)
- const std::string & **get** () const
- std::stringstream **getTextFormatted** () const override
- const std::string & **getId** () const
- void **accept** (Visitor *v) override

Additional Inherited Members

5.92.1 Member Function Documentation

5.92.1.1 accept()

```
void PlainSymbol::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

Reimplemented in [RuleWildcardExpression](#), and [VariableExpression](#).

5.92.1.2 getTextFormatted()

```
std::stringstream PlainSymbol::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Reimplemented from [Symbol](#).

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.93 point_t Struct Reference

Type for specifying position in input string.

```
#include <types.h>
```

Public Member Functions

- **point_t** (uint32_t row_pos, uint32_t col_pos)
- bool **operator<** (const [point_t](#) &r_point)
- bool **operator==** (const [point_t](#) &r_point)
- [point_t](#) **operator+** (const [point_t](#) &r_point)
- [point_t](#) **operator-** (const [point_t](#) &r_point)

Public Attributes

- uint32_t [row](#)
The row of input document.
- uint32_t [col](#)
The column of input document.

5.93.1 Detailed Description

Type for specifying position in input string.

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[types.h](#)

5.94 SymTabDuplId< V >::Pool Class Reference

[Pool](#) that owns all elements with the same id.

```
#include <syntab.h>
```

Classes

- class [iterator](#)
[Pool](#) iterator.

Public Types

- using **DataContainer** = std::vector< std::unique_ptr< V > >

Public Member Functions

- **Pool** (const [Pool](#) &)=delete
- [Pool](#) & **operator=** (const [Pool](#) &)=delete
- V * **first** () const
- size_t **size** () const
- bool **empty** () const
- V * **operator[]** (size_t i) const
- V * **at** (size_t i) const
- [iterator](#) **begin** () const
- [iterator](#) **end** () const
- DataContainer & **data** ()
- const DataContainer & **rd_data** () const
- void **erase** (V *item)

5.94.1 Detailed Description

```
template<class V>
class SymTabDuplId< V >::Pool
```

[Pool](#) that owns all elements with the same id.

5.94.2 Member Function Documentation

5.94.2.1 erase()

```
template<class V >
void SymTabDuplId< V >::Pool::erase (
    V * item )
```

< Related entry is found

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

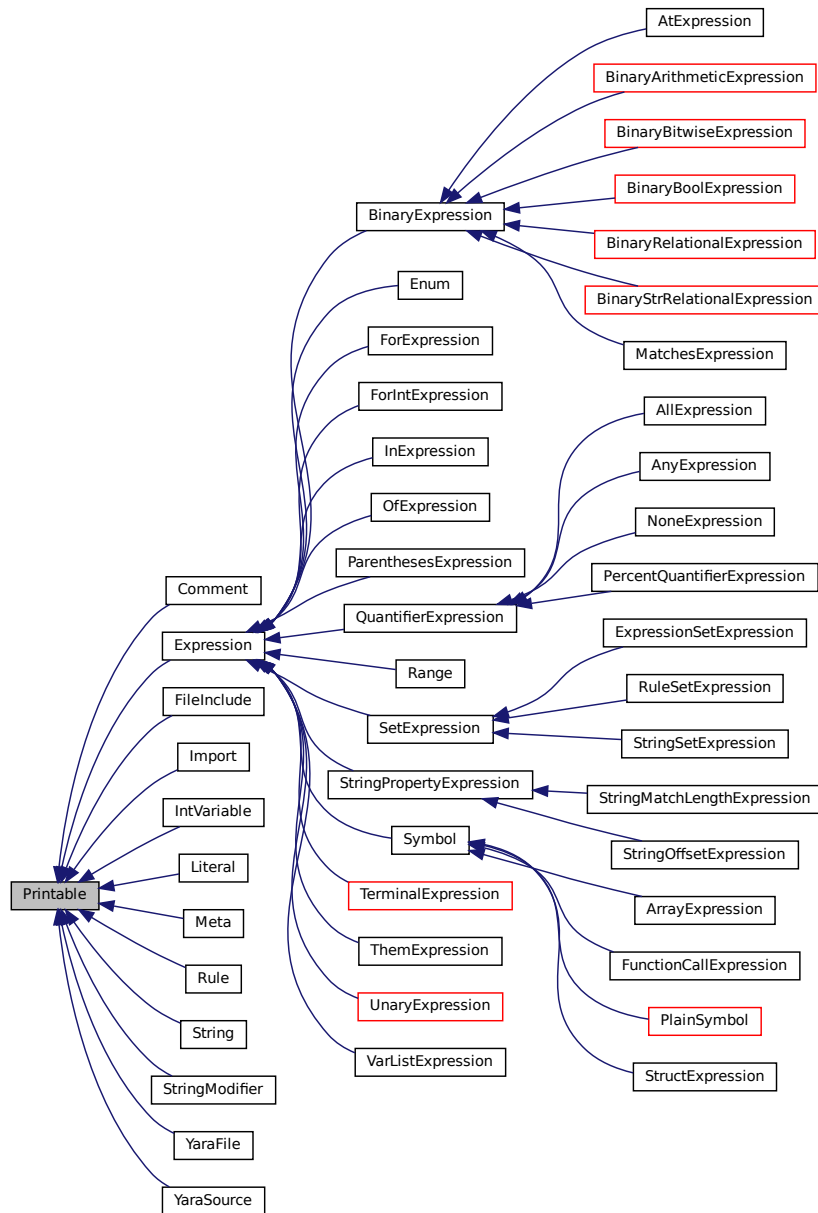
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[symtab.h](#)

5.95 Printable Class Reference

Base class for types, that can be serialized.

```
#include <common.h>
```

Inheritance diagram for Printable:



Public Member Functions

- virtual std::stringstream [getTextFormatted](#) () const =0

5.95.1 Detailed Description

Base class for types, that can be serialized.

5.95.2 Member Function Documentation

5.95.2.1 getTextFormatted()

```
virtual std::stringstream Printable::getTextFormatted ( ) const [pure virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

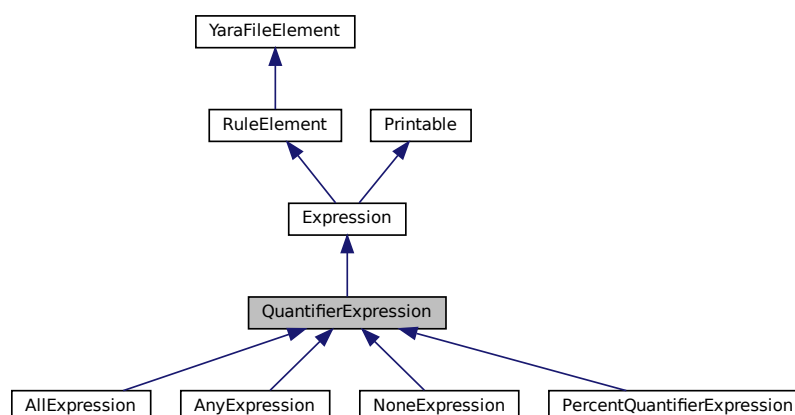
Implemented in [YaraFile](#), [IntVariable](#), [String](#), [StringModifier](#), [Import](#), [FileInclude](#), [Comment](#), [ForIntExpression](#), [VarListExpression](#), [ForExpression](#), [OfExpression](#), [ExpressionSetExpression](#), [RuleSetExpression](#), [StringSetExpression](#), [ThemExpression](#), [PercentQuantifierExpression](#), [AllExpression](#), [AnyExpression](#), [NoneExpression](#), [PlainSymbol](#), [FunctionCallExpression](#), [StructExpression](#), [ArrayExpression](#), [Symbol](#), [StringMatchLengthExpression](#), [StringOffsetExpression](#), [StringCountExpression](#), [StringExpression](#), [RegexExpression](#), [SizeExpression](#), [LiteralExpression](#), [UnaryExpression](#), [BinaryExpression](#), [InExpression](#), [ParenthesesExpression](#), [Enum](#), [Range](#), [YaraSource](#), [Rule](#), [Meta](#), and [Literal](#).

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

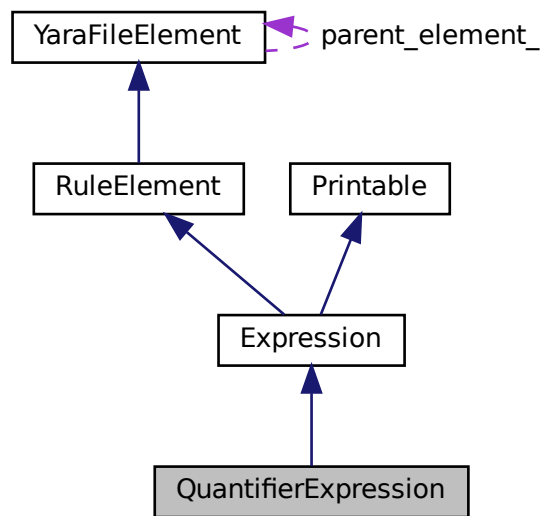
- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/common.h`

5.96 QuantifierExpression Class Reference

Inheritance diagram for QuantifierExpression:



Collaboration diagram for QuantifierExpression:



Public Member Functions

- bool [isValid](#) (std::string_view &) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- [Expression::Type](#) [getType](#) () const override

Additional Inherited Members

5.96.1 Member Function Documentation

5.96.1.1 areOperandsValid()

```
bool QuantifierExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

Reimplemented in [PercentQuantifierExpression](#).

5.96.1.2 getType()

```
Expression::Type QuantifierExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

Reimplemented in [PercentQuantifierExpression](#).

5.96.1.3 isComplete()

```
bool QuantifierExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

Reimplemented in [PercentQuantifierExpression](#).

5.96.1.4 isValid()

```
bool QuantifierExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

Reimplemented in [PercentQuantifierExpression](#).

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

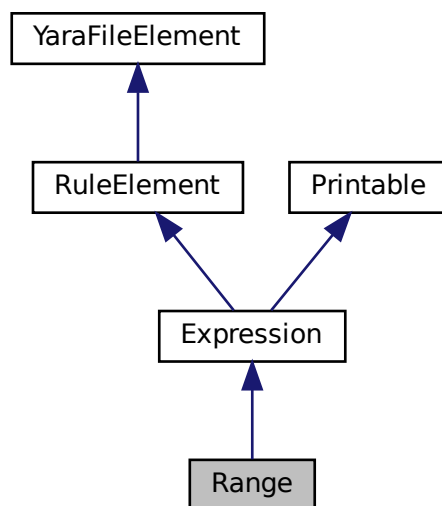
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.97 Range Class Reference

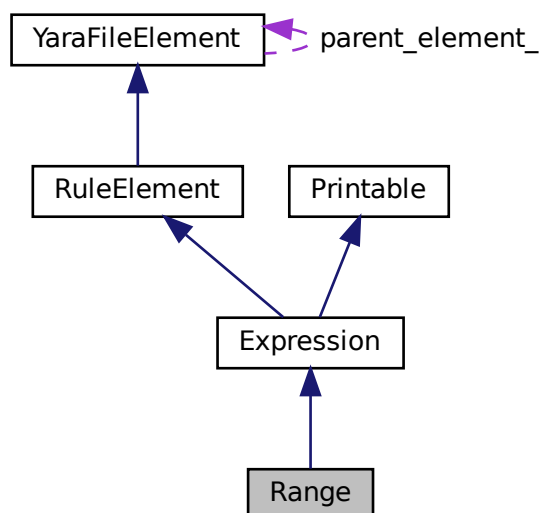
Class, that represents integer range (for example "(1..200)")

```
#include <expression.h>
```

Inheritance diagram for Range:



Collaboration diagram for Range:



Public Member Functions

- **Range** (ExpressionPtr from, ExpressionPtr to)
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override

Provides way how to determine if range as all necessary members defined.
- bool [isValid](#) (std::string_view &msg) const override

Performs check if range is valid (both integer expressions must be valid as well as range itself)
- [Expression::Type](#) [getType](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) (Visitor *v) override
- const ExpressionPtr & [getFrom](#) () const
- const ExpressionPtr & [getTo](#) () const

Additional Inherited Members

5.97.1 Detailed Description

Class, that represents integer range (for example "(1..200)")

5.97.2 Member Function Documentation

5.97.2.1 [accept\(\)](#)

```
void Range::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.97.2.2 [areOperandsValid\(\)](#)

```
bool Range::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.97.2.3 getTextFormatted()

```
std::stringstream Range::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.97.2.4 getType()

```
Expression::Type Range::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

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

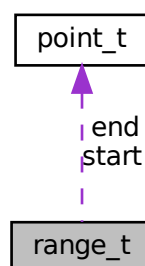
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.98 range_t Struct Reference

[Range](#) in input document specified by start point and end point.

```
#include <types.h>
```

Collaboration diagram for range_t:



Public Member Functions

- **range_t** ([point_t](#) start_pt, [point_t](#) end_pt)
- **range_t** (uint32_t start_row, uint32_t start_col, uint32_t end_row, uint32_t end_col)
- **range_t operator+** (const [point_t](#) &r_point)
- **range_t operator-** (const [point_t](#) &r_point)

Public Attributes

- [point_t](#) start
Start point of range (included in range)
- [point_t](#) end
End point of range (NOT included in range)

5.98.1 Detailed Description

[Range](#) in input document specified by start point and end point.

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

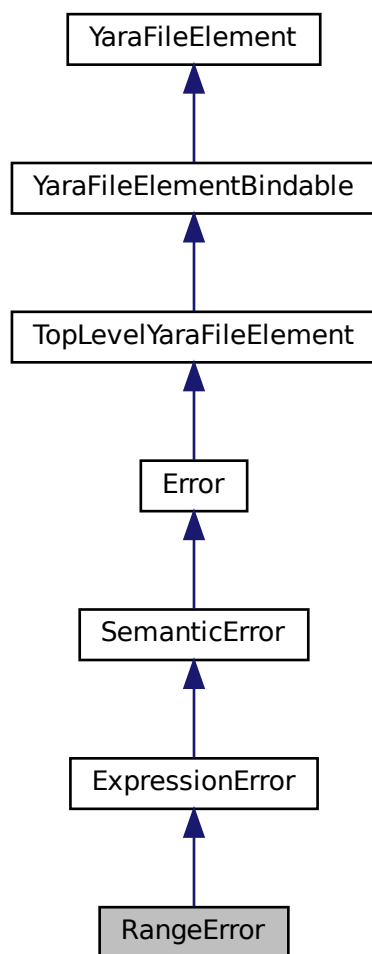
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[types.h](#)

5.99 RangeError Class Reference

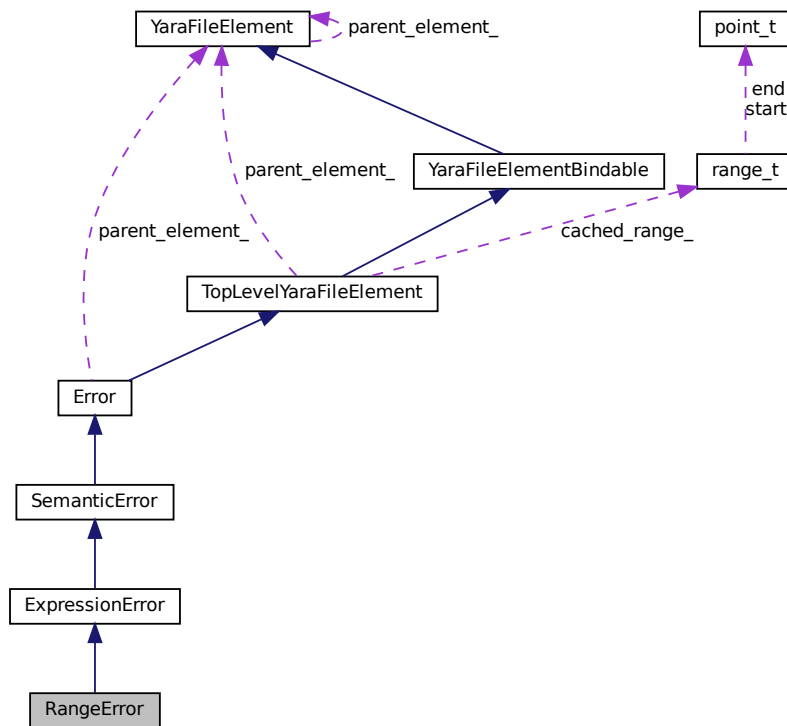
Bad range (e. g. left bound is greater than right bound)

```
#include <error.h>
```

Inheritance diagram for RangeError:



Collaboration diagram for RangeError:



Public Member Functions

- **RangeError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [RangeErrorException](#) exception ()

Additional Inherited Members

5.99.1 Detailed Description

Bad range (e. g. left bound is greater than right bound)

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

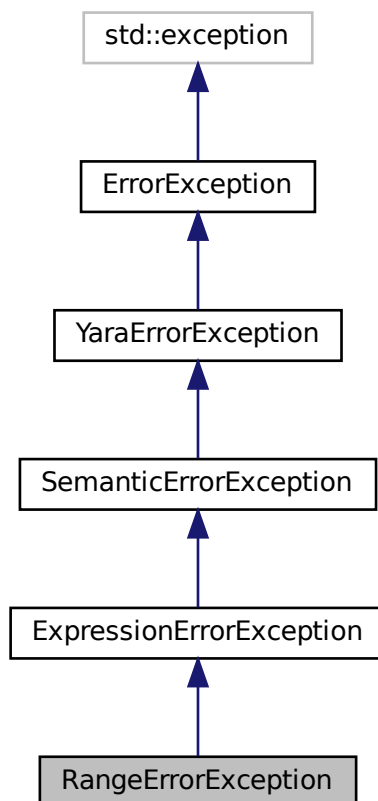
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.100 RangeErrorException Class Reference

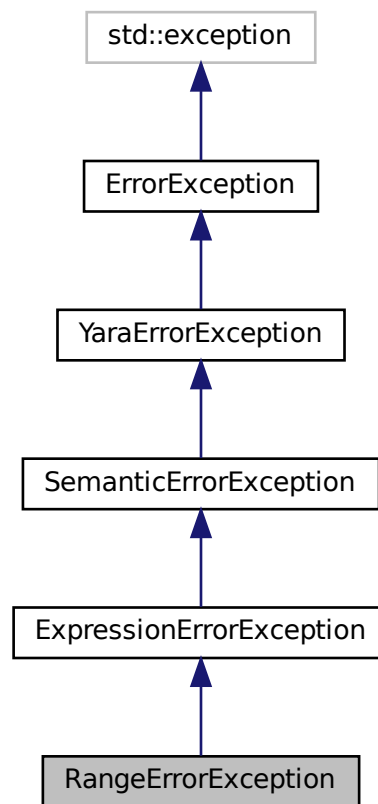
Exception for errors, that occur in range expression (e. g. higher left bound than right bound)

```
#include <error.h>
```

Inheritance diagram for RangeErrorException:



Collaboration diagram for RangeErrorException:



Public Member Functions

- **RangeErrorException** (offset_t offset, size_t len, std::string msg={})

5.100.1 Detailed Description

Exception for errors, that occur in range expression (e. g. higher left bound than right bound)

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

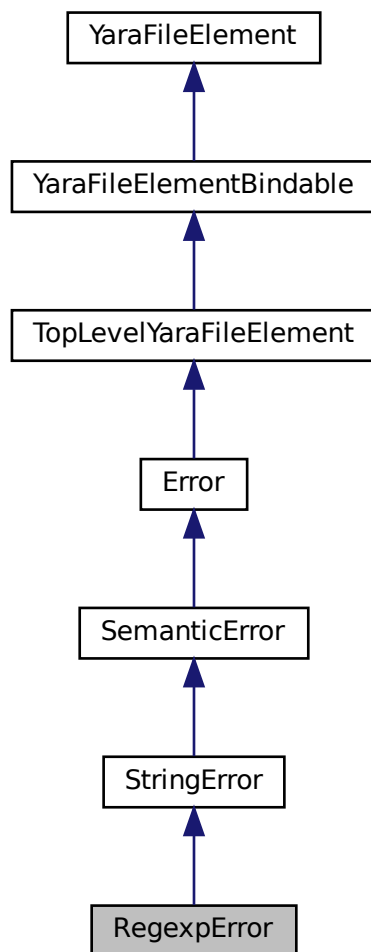
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.101 RegexpError Class Reference

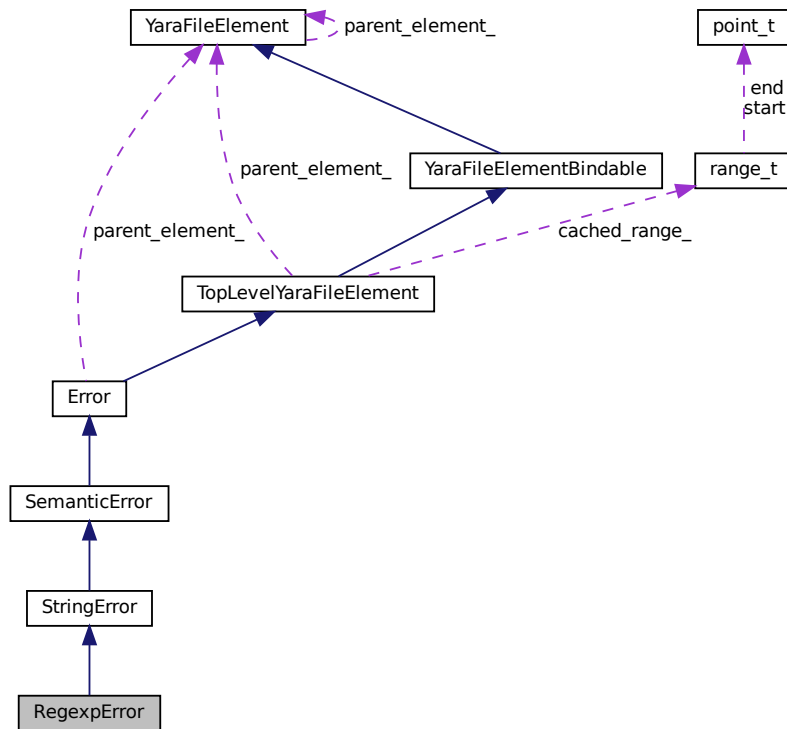
Regular expression error (e. g. bad repeat interval)

```
#include <error.h>
```

Inheritance diagram for RegexpError:



Collaboration diagram for RegexpError:



Public Member Functions

- **RegexpError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [RegexpErrorException](#) exception ()

Additional Inherited Members

5.101.1 Detailed Description

Regular expression error (e. g. bad repeat interval)

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

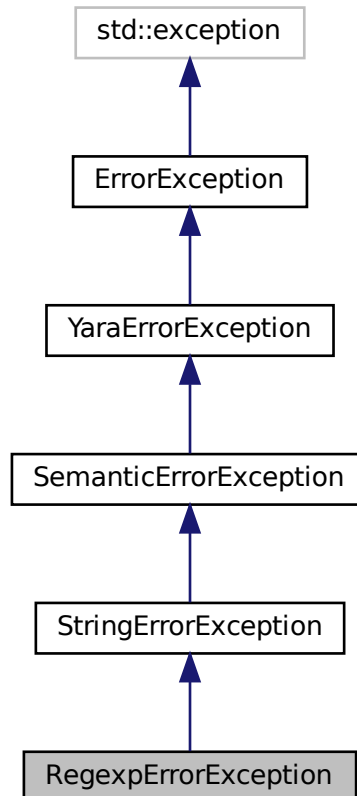
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.102 RegexpErrorException Class Reference

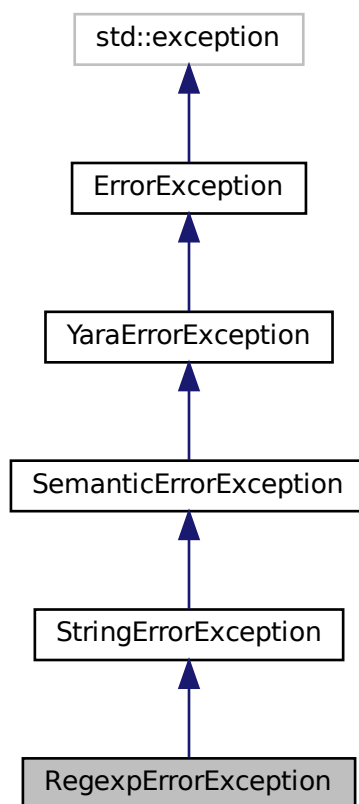
Exception of semantic error of regular expression (e. g. bad repeat interval)

```
#include <error.h>
```

Inheritance diagram for RegexpErrorException:



Collaboration diagram for RegexpErrorException:



Public Member Functions

- **RegexpErrorException** (offset_t offset, size_t len, std::string msg={})

5.102.1 Detailed Description

Exception of semantic error of regular expression (e. g. bad repeat interval)

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

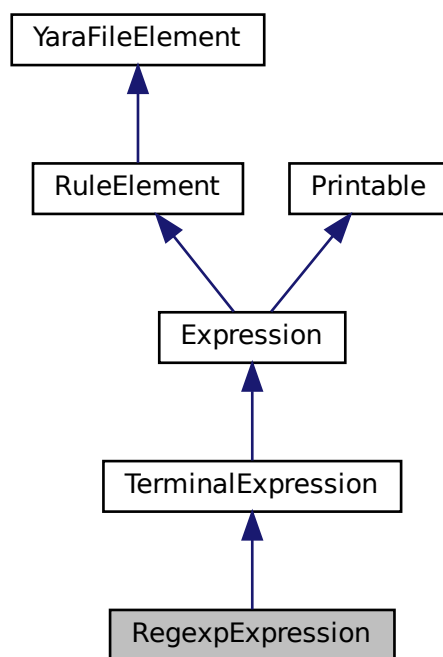
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.103 RegexpExpression Class Reference

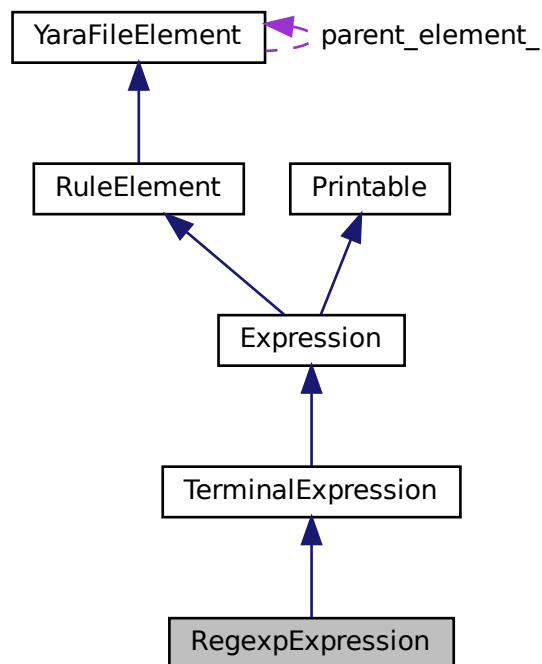
Represents regular expression literal in expressions.

```
#include <expression.h>
```

Inheritance diagram for RegexpExpression:



Collaboration diagram for RegexpExpression:



Public Member Functions

- **RegexpExpression** (const std::string &content)
- bool **isComplete** () const override
- **Expression::Type** **getType** () const override
- std::stringstream **getTextFormatted** () const override
- void **accept** (Visitor *v) override
- const std::string & **getContent** () const

Additional Inherited Members

5.103.1 Detailed Description

Represents regular expression literal in expressions.

5.103.2 Member Function Documentation

5.103.2.1 accept()

```
void RegexpExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.103.2.2 getTextFormatted()

```
std::stringstream RegexpExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.103.2.3 getType()

```
Expression::Type RegexpExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.103.2.4 isComplete()

```
bool RegexpExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

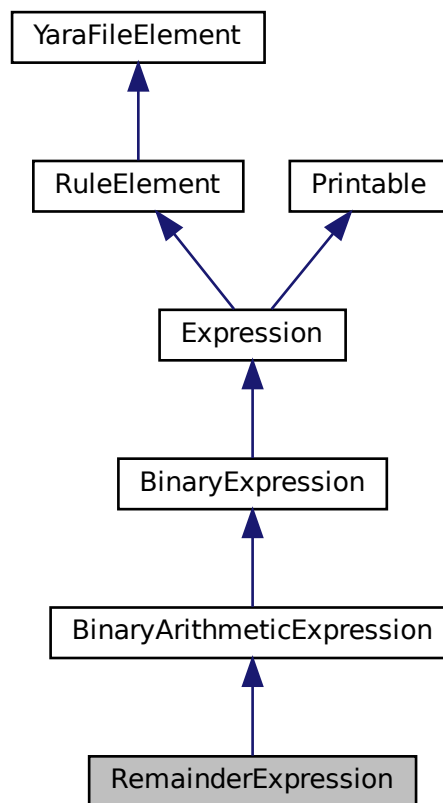
Implements [Expression](#).

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

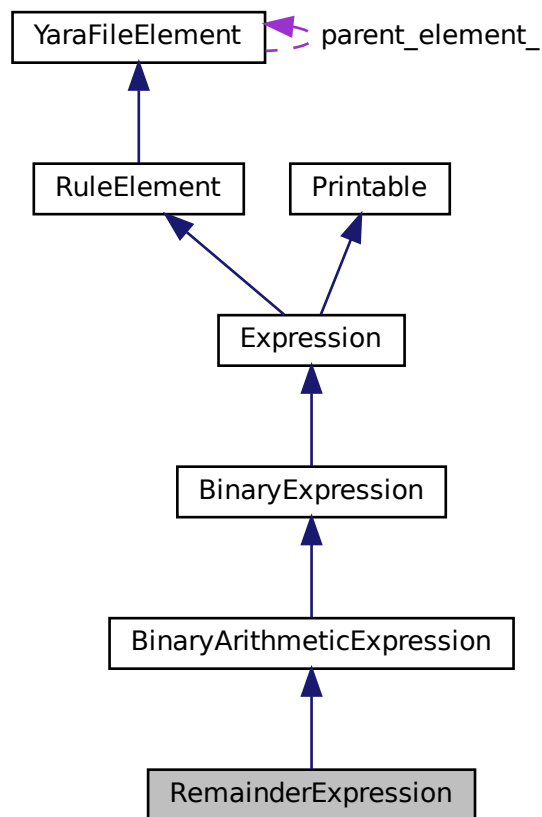
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.104 RemainderExpression Class Reference

Inheritance diagram for RemainderExpression:



Collaboration diagram for RemainderExpression:



Public Member Functions

- **RemainderExpression** (ExpressionPtr lop, ExpressionPtr rop)
- `bool isValid (std::string_view &msg) const` override
- `Expression::Type getType ()` const override
- `std::string opsign ()` const override
- `void accept (Visitor *v)` override

Additional Inherited Members

5.104.1 Member Function Documentation

5.104.1.1 accept()

```
void RemainderExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.104.1.2 getType()

```
Expression::Type RemainderExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Reimplemented from [BinaryArithmeticExpression](#).

5.104.1.3 isValid()

```
bool RemainderExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Reimplemented from [BinaryArithmeticExpression](#).

5.104.1.4 opsign()

```
std::string RemainderExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

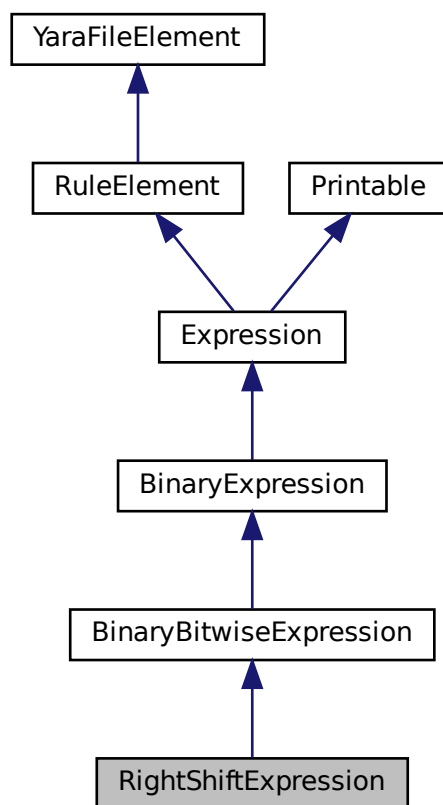
Implements [BinaryExpression](#).

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

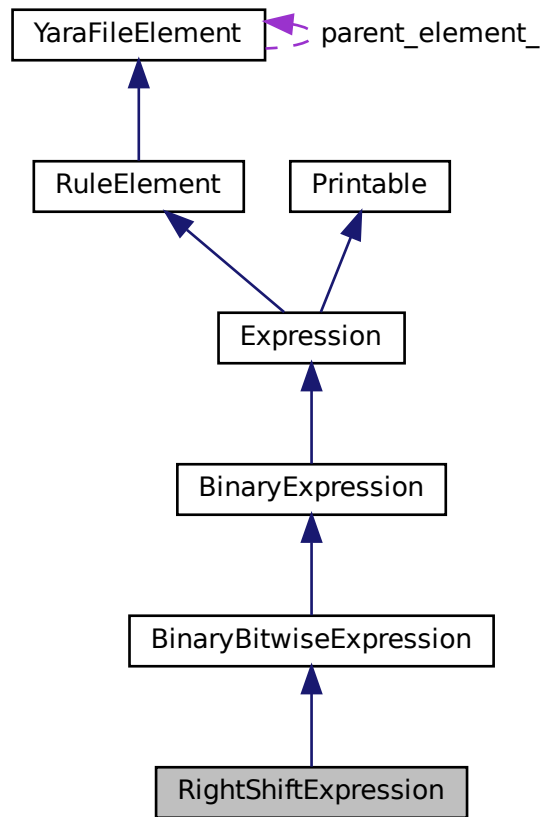
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.105 RightShiftExpression Class Reference

Inheritance diagram for RightShiftExpression:



Collaboration diagram for RightShiftExpression:



Public Member Functions

- **RightShiftExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string `opsign` () const override
- void `accept` (Visitor *v) override

Additional Inherited Members

5.105.1 Member Function Documentation

5.105.1.1 `accept()`

```
void RightShiftExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.105.1.2 opsign()

```
std::string RightShiftExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

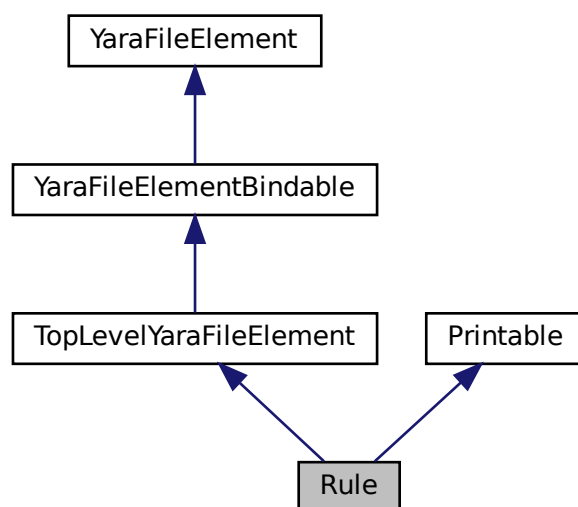
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.106 Rule Class Reference

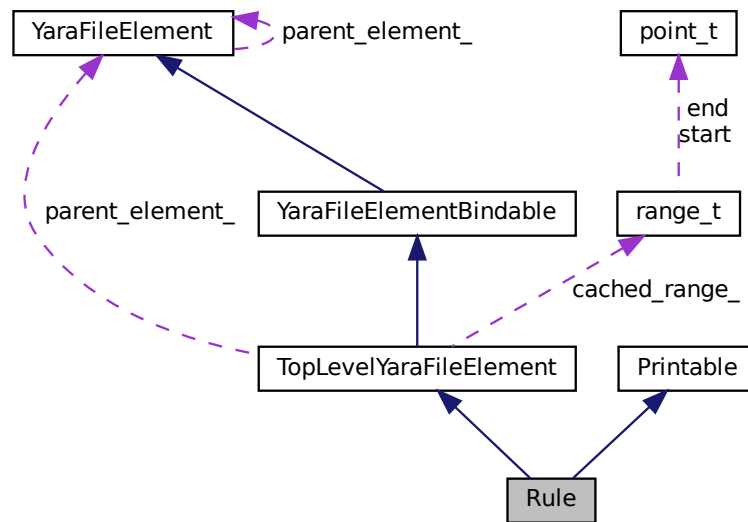
Class that represents yara rule.

```
#include <rule.h>
```

Inheritance diagram for Rule:



Collaboration diagram for Rule:



Public Member Functions

- **Rule** (const [Rule](#) &)=delete
- **Rule** & **operator=** (const [Rule](#) &)=delete
- const std::string & **getId** () const
- void **setId** (const std::string &id)
- const std::vector< std::string > & **getTags** () const
- void **addTag** (const std::string &new_tag)
- bool **hasTag** (const std::string &tag)
- void **removeTag** (const std::string &tag)
- const [RuleModifierContainer](#) & **getModifiers** () const
- void **addModifier** (std::unique_ptr< [RuleModifier](#) > &&new_mod)
- void **removeModifier** ([RuleModifier::Type](#) type)
- bool **isGlobal** ()
- bool **isPrivate** ()
- const [SymTabDuplId](#)< [Meta](#) > & **getMetas** () const
- const [SymTabDuplId](#)< [Meta](#) >::Pool & **getMetasById** (const std::string &id) const
- void **addMeta** (std::unique_ptr< [Meta](#) > &&new_meta)
- void **removeMeta** (const std::string &id)
- const [SymTab](#)< [String](#) > & **getStrings** () const
- const std::unique_ptr< [String](#) > & **getStringById** (const std::string &id) const
- void **addString** (std::unique_ptr< [String](#) > &&new_string)
- void **removeString** (const std::string &id)
- const [SymTab](#)< [IntVariable](#) > & **getVars** () const
- const std::unique_ptr< [IntVariable](#) > & **getVarById** (const std::string &id) const
- void **addVar** (std::unique_ptr< [IntVariable](#) > &&new_var)
- void **removeVar** (const std::string &id)
- void **setCondition** (ExpressionPtr &&condition_expr)
- void **setCondition** (const ExpressionPtr &condition_expr)

- const ExpressionPtr & [getCondition](#) () const
- offset_t [getIdOffset](#) () const
- void [setIdOffset](#) (const offset_t &offset)
- std::stringstream [getTextFormatted](#) () const

Additional Inherited Members

5.106.1 Detailed Description

Class that represents yara rule.

Represents the whole sequence of tokens, that starts with rule modifier and ends with '}'

5.106.2 Member Function Documentation

5.106.2.1 addMeta()

```
void Rule::addMeta (
    std::unique_ptr< Meta > && new_meta )
```

Adds meta to symbol tab of this rule

5.106.2.2 addModifier()

```
void Rule::addModifier (
    std::unique_ptr< RuleModifier > && new_mod )
```

Adds modifier to the rule

Note

If rule already has the modifier with the same type exception is thrown

< Modifier is already in the modifier list

5.106.2.3 addString()

```
void Rule::addString (
    std::unique_ptr< String > && new_string )
```

Adds string to the rule < If string is anonymous, generate unique ID for storing it in symbol table

< [String](#) with this id is already declared

< Increment length by 1 because dollar sign is shifted from Id

5.106.2.4 addTag()

```
void Rule::addTag (
    const std::string & new_tag )
```

Adds tag to the rule tags < Tag is already in tag list

5.106.2.5 addVar()

```
void Rule::addVar (
    std::unique_ptr< IntVariable > && new_var )
```

Adds the new internal variable < Check if there is already variable with this id in symbol tab

5.106.2.6 getCondition()

```
const ExpressionPtr & Rule::getCondition ( ) const
```

Returns reference to the condition expression

5.106.2.7 getId()

```
const std::string & Rule::getId ( ) const
```

Returns rule id

Returns

const reference to string with rule id

5.106.2.8 getIdOffset()

```
offset_t Rule::getIdOffset ( ) const
```

Returns global offset of rule id

5.106.2.9 getMetas()

```
const SymTabDuplId< Meta > & Rule::getMetas ( ) const
```

Returns all metas (symtab, that contains them) of the rule

5.106.2.10 getMetasById()

```
const SymTabDuplId< Meta >::Pool & Rule::getMetasById (
    const std::string & id ) const
```

Returns pool with metas, that have the same id, that is given as parameter of this method

5.106.2.11 getModifiers()

```
const RuleModifierContainer & Rule::getModifiers ( ) const
```

Returns all rule modifiers

5.106.2.12 getStringById()

```
const std::unique_ptr< String > & Rule::getStringById (
    const std::string & id ) const
```

Returns rule string, that has given id

5.106.2.13 getStrings()

```
const SymTab< String > & Rule::getStrings ( ) const
```

Returns reference to symtab with all strings of the rule

5.106.2.14 getTags()

```
const std::vector< std::string > & Rule::getTags ( ) const
```

Get all tags of the rule

Returns

unordered set with tags

5.106.2.15 getTextFormatted()

```
std::stringstream Rule::getTextFormatted ( ) const [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.106.2.16 getVarById()

```
const std::unique_ptr< IntVariable > & Rule::getVarById (
    const std::string & id ) const
```

Returns reference to specific internal variable

5.106.2.17 getVars()

```
const SymTab< IntVariable > & Rule::getVars ( ) const
```

Returns symbol tab with all internal variables of the rule

Note

internal variables are Avast feature of YARA

5.106.2.18 hasTag()

```
bool Rule::hasTag (
    const std::string & tag )
```

Checks whether the rule has given tag

5.106.2.19 isGlobal()

```
bool Rule::isGlobal ( )
```

Check whether the rule has global modifier

5.106.2.20 isPrivate()

```
bool Rule::isPrivate ( )
```

Check whether the rule has private modifier

5.106.2.21 removeMeta()

```
void Rule::removeMeta (
    const std::string & id )
```

Removes all metas with given id from the symbol tab

5.106.2.22 removeModifier()

```
void Rule::removeModifier (
    RuleModifier::Type type )
```

Removes modifier with given type from the rule modifiers

5.106.2.23 removeString()

```
void Rule::removeString (
    const std::string & id )
```

Removes string from the symtab of rule strings

5.106.2.24 removeTag()

```
void Rule::removeTag (
    const std::string & tag )
```

Removes tag from the rule tags

5.106.2.25 removeVar()

```
void Rule::removeVar (
    const std::string & id )
```

Removes internal variable with given id from the symtab

5.106.2.26 setCondition()

```
void Rule::setCondition (
    ExpressionPtr && condition_expr )
```

Sets the condition expression

5.106.2.27 setId()

```
void Rule::setId (
    const std::string & id )
```

Sets the id of the rule

5.106.2.28 setIdOffset()

```
void Rule::setIdOffset (
    const offset_t & offset )
```

Sets the global offset of the rule id

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

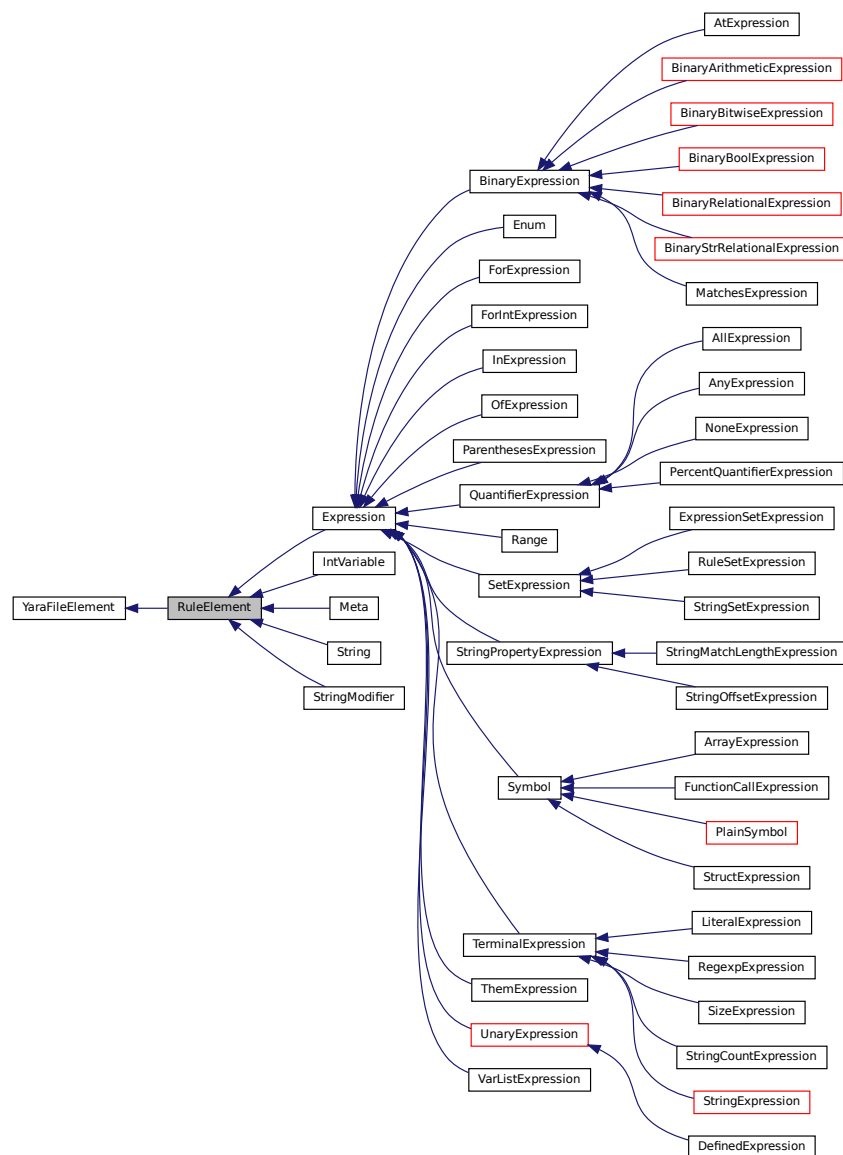
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/rule.cpp](#)

5.107 RuleElement Class Reference

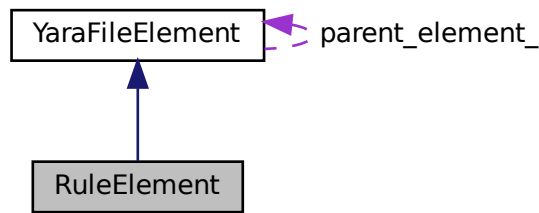
Base class for all symbols that are valid in rule context.

```
#include <rule_element.h>
```

Inheritance diagram for RuleElement:



Collaboration diagram for RuleElement:



Public Member Functions

- `offset_t` [getOffset](#) () const override
- void [setOffset](#) (const `offset_t` &offset) override
- `offset_t` [getLocalOffset](#) ()
- void [setLocalOffset](#) (const `offset_t` &offset)
- `range_t` [getRange](#) () const
- void [setRange](#) (const `range_t` &range)
- `Rule *` [getParentRule](#) () const
- void [setParentRule](#) (const `RulePtr` &parent_rule)
- void [setParentRule](#) (`Rule *`parent_rule)

Additional Inherited Members

5.107.1 Detailed Description

Base class for all symbols that are valid in rule context.

5.107.2 Member Function Documentation

5.107.2.1 `getLocalOffset()`

```
offset_t RuleElement::getLocalOffset ( )
```

Returns local offset of rule element (can be computed as 'global offset of rule element' - 'global offset of parent rule')

5.107.2.2 getOffset()

```
offset_t RuleElement::getOffset ( ) const [override], [virtual]
```

Returns global offset (offset in yara file) of rule element < Convert local offset to global offset by adding rule offset

Reimplemented from [YaraFileElement](#).

5.107.2.3 getParentRule()

```
Rule * RuleElement::getParentRule ( ) const
```

Returns pointer to parent rule of rule element

5.107.2.4 getRange()

```
range_t RuleElement::getRange ( ) const
```

Returns [Range](#) where is [RuleElement](#) located (it is global location)

5.107.2.5 setLocalOffset()

```
void RuleElement::setLocalOffset (
    const offset_t & offset )
```

Sets local offset of rule element

5.107.2.6 setOffset()

```
void RuleElement::setOffset (
    const offset_t & offset ) [override], [virtual]
```

Sets global offset of rule element < Only local offset is stored

Reimplemented from [YaraFileElement](#).

5.107.2.7 setParentRule()

```
void RuleElement::setParentRule (
    const RulePtr & parent_rule )
```

Sets parent rule of the rule element

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

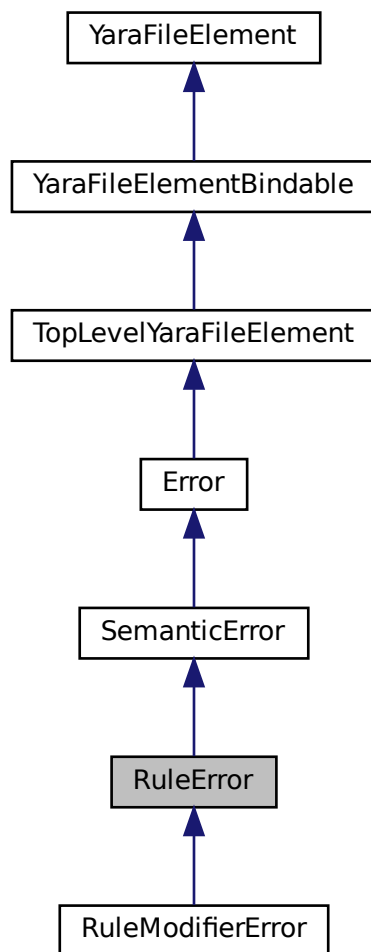
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[rule_element.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[rule_element.cpp](#)

5.108 RuleError Class Reference

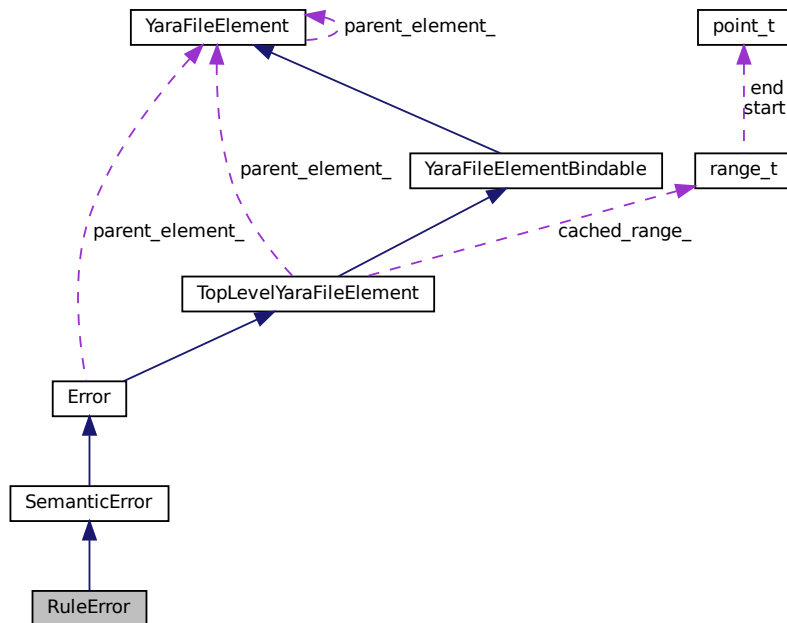
[Error](#), that occurred in error definition (e. g. collision of rule identifiers)

```
#include <error.h>
```

Inheritance diagram for RuleError:



Collaboration diagram for RuleError:



Public Member Functions

- **RuleError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [RuleErrorException](#) exception ()

Additional Inherited Members

5.108.1 Detailed Description

[Error](#), that occurred in error definition (e. g. collision of rule identifiers)

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

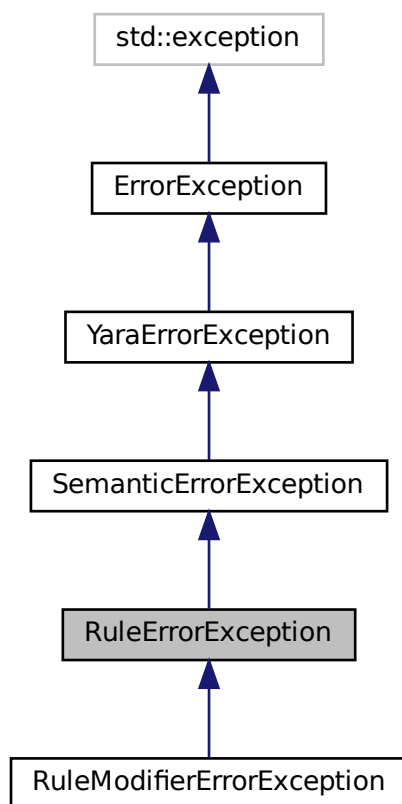
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.109 RuleErrorException Class Reference

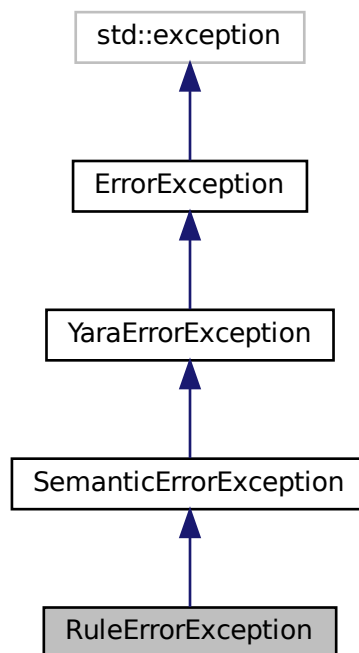
Exception for errors occurred in rule definition (e. g. collision of rule identifier)

```
#include <error.h>
```

Inheritance diagram for RuleErrorException:



Collaboration diagram for RuleErrorException:



Public Member Functions

- **RuleErrorException** (offset_t offset, size_t len, std::string msg={})

5.109.1 Detailed Description

Exception for errors occurred in rule definition (e. g. collision of rule identifier)

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

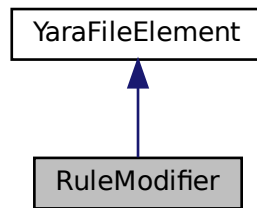
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.110 RuleModifier Class Reference

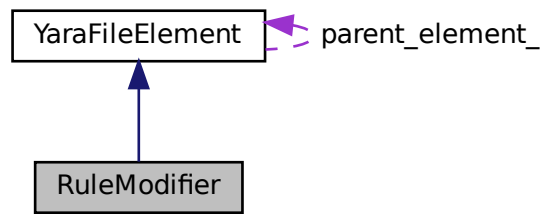
Class that represents rule modifiers.

```
#include <rule.h>
```

Inheritance diagram for RuleModifier:



Collaboration diagram for RuleModifier:



Public Types

- enum class [Type](#) { [Private](#) , [Global](#) }
- Type of rule the rule modifier.*

Public Member Functions

- **RuleModifier** ([RuleModifier::Type](#) type)
- [RuleModifier::Type](#) [getType](#) ()

Static Public Member Functions

- static [Type](#) [stringToType](#) (std::string_view str)
- static std::string_view [typeToString](#) ([Type](#) type)

Additional Inherited Members

5.110.1 Detailed Description

Class that represents rule modifiers.

5.110.2 Member Enumeration Documentation

5.110.2.1 Type

```
enum RuleModifier::Type [strong]
```

Type of rule the rule modifier.

Enumerator

Private	Private rule.
Global	Global rule.

5.110.3 Member Function Documentation

5.110.3.1 getType()

```
RuleModifier::Type RuleModifier::getType ( )
```

Returns type of the rule modifier

5.110.3.2 stringToType()

```
RuleModifier::Type RuleModifier::stringToType (
    std::string_view str ) [static]
```

Provides conversion from string view to [RuleModifier::Type](#)

5.110.3.3 typeToString()

```
std::string_view RuleModifier::typeToString (
    RuleModifier::Type type ) [static]
```

Provides conversion [RuleModifier::Type](#) to string view

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[rule.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/rule.cpp

5.111 RuleModifierContainer Class Reference

```
#include <rule.h>
```

Classes

- class [iterator](#)
Iterator for [RuleModifierContainer](#).

Public Member Functions

- **RuleModifierContainer** (const [RuleModifierContainer](#) &)=delete
- **RuleModifierContainer** & **operator=** (const [RuleModifierContainer](#) &)=delete
- size_t **size** () const
- bool **empty** () const
- bool **has** ([RuleModifier::Type](#) type) const
- bool **hasGlobal** () const
- bool **hasPrivate** () const
- [RuleModifier](#) * **getGlobal** () const
- [RuleModifier](#) * **getPrivate** () const
- [RuleModifierContainer::iterator](#) **begin** () const
- [RuleModifierContainer::iterator](#) **end** () const
- void **add** (std::unique_ptr< [RuleModifier](#) > &&new_mod)
- void **erase** ([RuleModifier::Type](#) type)

5.111.1 Detailed Description

Container for rule modifiers of rule. Because in YARA rule can have only two modifiers (global and private), this container should be sufficient.

5.111.2 Member Function Documentation

5.111.2.1 add()

```
void RuleModifierContainer::add (
    std::unique_ptr< RuleModifier > && new_mod )
```

Adds modifier to the container

5.111.2.2 begin()

```
RuleModifierContainer::iterator RuleModifierContainer::begin ( ) const
```

Returns begin iterator of container

5.111.2.3 empty()

```
bool RuleModifierContainer::empty ( ) const
```

Checks whether there is at least one modifier in container

5.111.2.4 end()

```
RuleModifierContainer::iterator RuleModifierContainer::end ( ) const
```

Returns end iterator of container

5.111.2.5 erase()

```
void RuleModifierContainer::erase (
    RuleModifier::Type type )
```

Erases modifier with given type from the container

Note

It does nothing, if there is no such modifier

5.111.2.6 getGlobal()

```
RuleModifier * RuleModifierContainer::getGlobal ( ) const
```

Returns pointer to global modifier

5.111.2.7 getPrivate()

```
RuleModifier * RuleModifierContainer::getPrivate ( ) const
```

Returns pointer to private modifier

5.111.2.8 has()

```
bool RuleModifierContainer::has (
    RuleModifier::Type type ) const
```

Checks if container contains modifier with given type

Returns

true if container contains modifier with given type

5.111.2.9 hasGlobal()

```
bool RuleModifierContainer::hasGlobal ( ) const
```

Checks if container contains global modifier

5.111.2.10 hasPrivate()

```
bool RuleModifierContainer::hasPrivate ( ) const
```

Checks if container contains private modifier

5.111.2.11 size()

```
size_t RuleModifierContainer::size ( ) const
```

Returns the number of rule modifiers

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

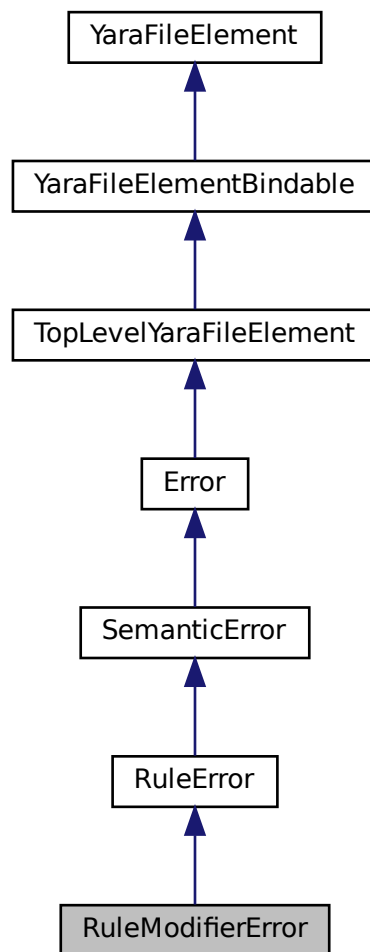
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/rule.cpp](#)

5.112 RuleModifierError Class Reference

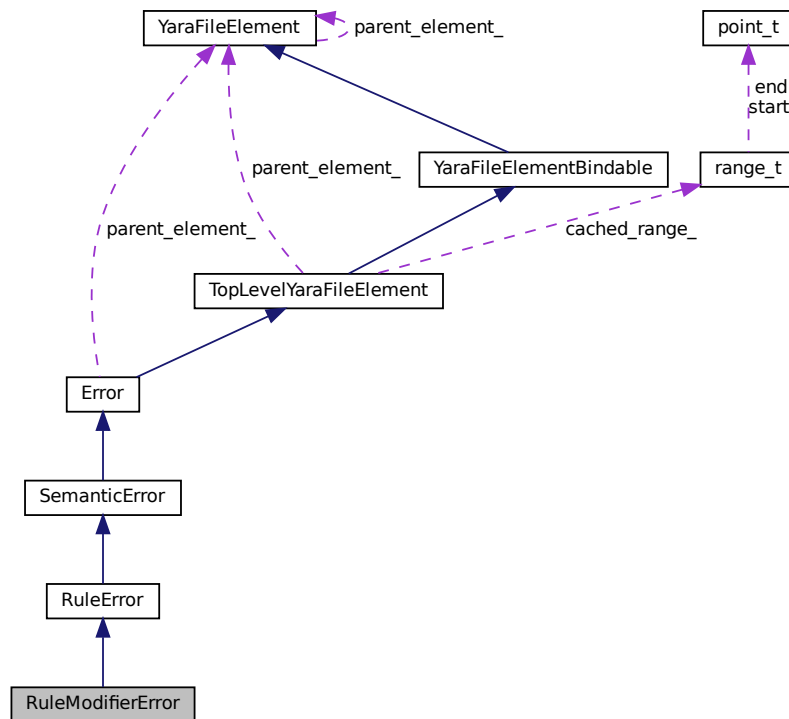
[Error](#) of rule modifier (e. g. duplicated rule modifier)

```
#include <error.h>
```

Inheritance diagram for RuleModifierError:



Collaboration diagram for RuleModifierError:



Public Member Functions

- **RuleModifierError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [RuleModifierErrorException](#) exception ()

Additional Inherited Members

5.112.1 Detailed Description

[Error](#) of rule modifier (e. g. duplicated rule modifier)

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

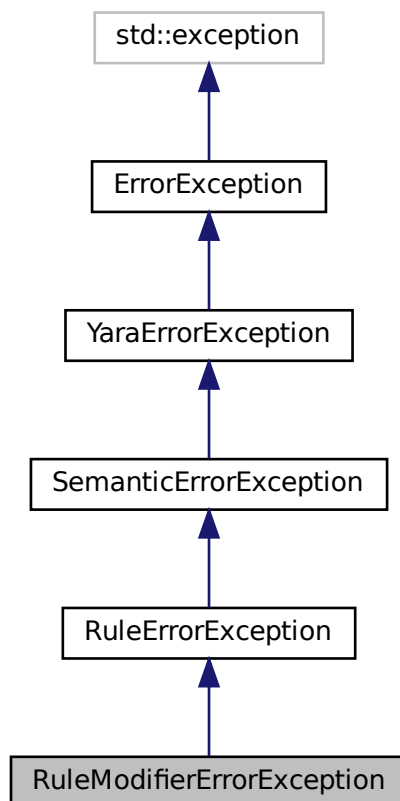
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.113 RuleModifierErrorException Class Reference

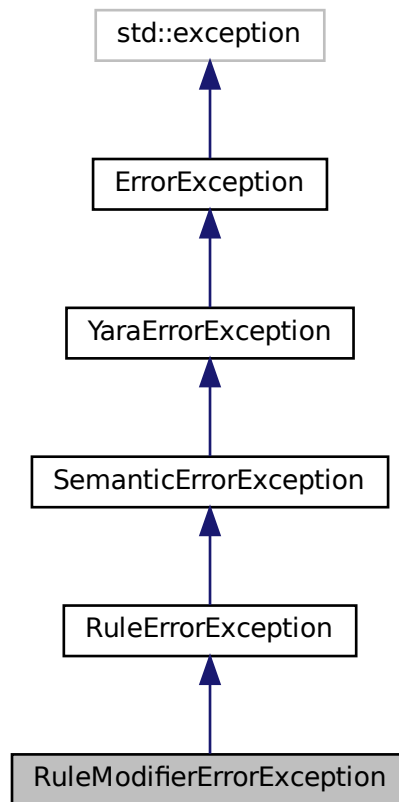
Exception for errors of rule modifier (e. g. duplicated rule modifier)

```
#include <error.h>
```

Inheritance diagram for RuleModifierErrorException:



Collaboration diagram for RuleModifierErrorException:



Public Member Functions

- **RuleModifierErrorException** (offset_t offset, size_t len, std::string msg={})

5.113.1 Detailed Description

Exception for errors of rule modifier (e. g. duplicated rule modifier)

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

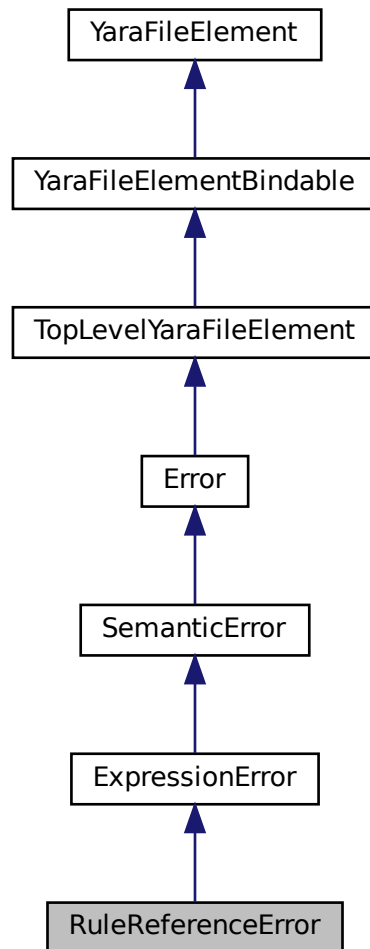
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.114 RuleReferenceError Class Reference

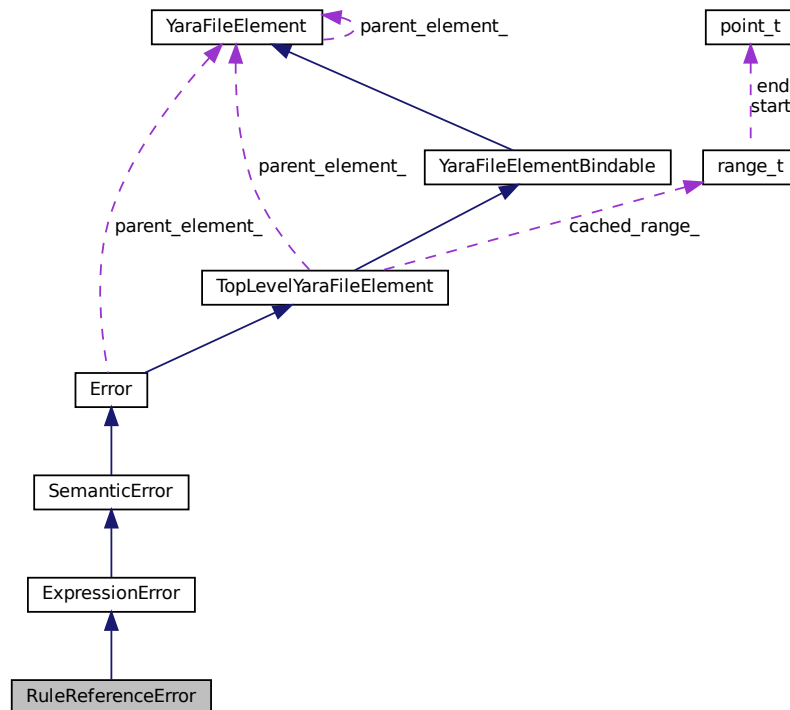
Bad rule reference (referenced rule does not exists)

```
#include <error.h>
```

Inheritance diagram for RuleReferenceError:



Collaboration diagram for RuleReferenceError:



Public Member Functions

- **RuleReferenceError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [RuleReferenceErrorException](#) exception ()

Additional Inherited Members

5.114.1 Detailed Description

Bad rule reference (referenced rule does not exists)

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

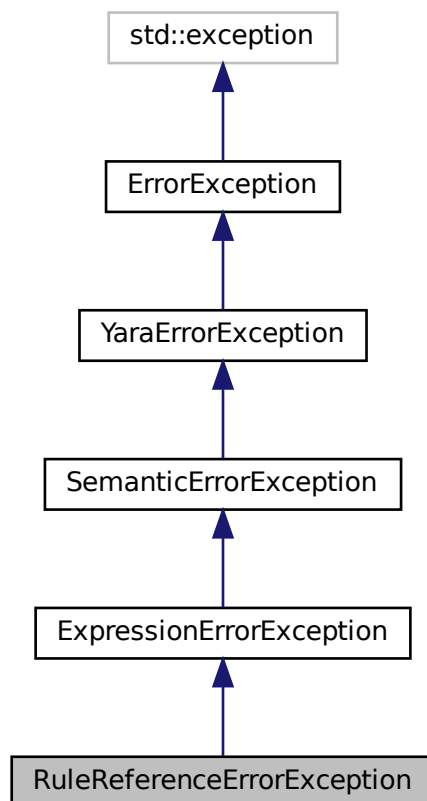
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.115 RuleReferenceErrorException Class Reference

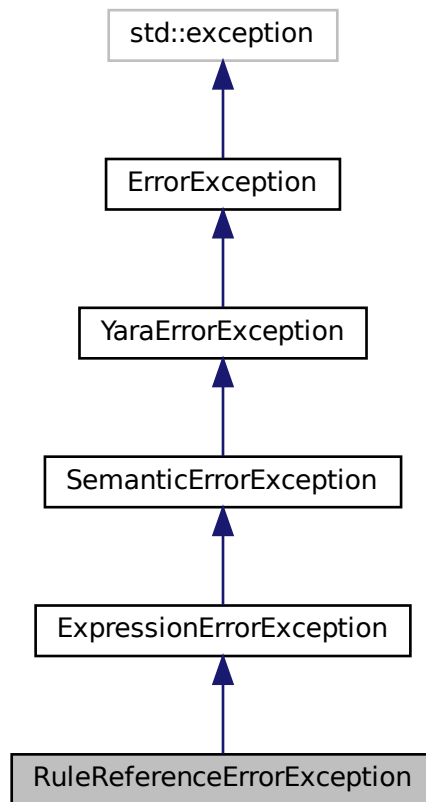
Exception for bad rule reference (e. g. rule does not exists)

```
#include <error.h>
```

Inheritance diagram for RuleReferenceErrorException:



Collaboration diagram for RuleReferenceErrorException:



Public Member Functions

- **RuleReferenceErrorException** (offset_t offset, size_t len, std::string msg={})

5.115.1 Detailed Description

Exception for bad rule reference (e. g. rule does not exists)

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

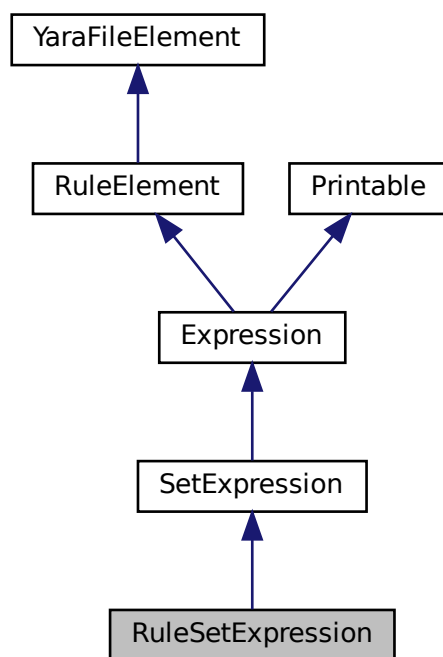
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.116 RuleSetExpression Class Reference

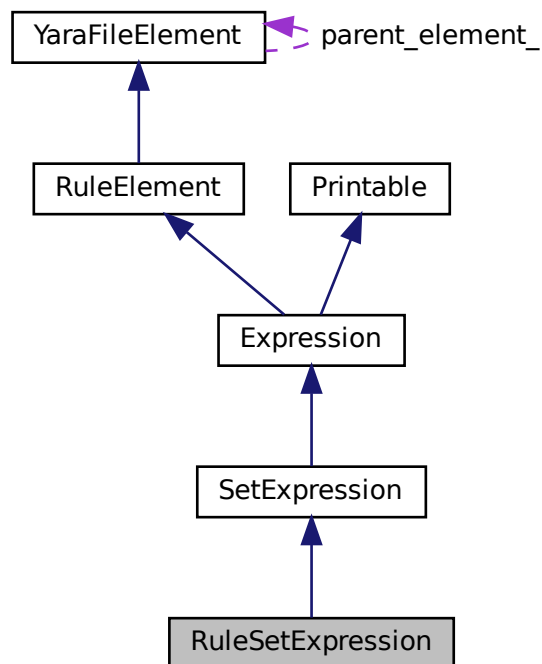
Represents set of rules.

```
#include <expression.h>
```

Inheritance diagram for RuleSetExpression:



Collaboration diagram for RuleSetExpression:



Public Member Functions

- **RuleSetExpression** (std::vector< ExpressionPtr > elements)
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.116.1 Detailed Description

Represents set of rules.

5.116.2 Member Function Documentation

5.116.2.1 accept()

```
void RuleSetExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.116.2.2 getTextFormatted()

```
std::stringstream RuleSetExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

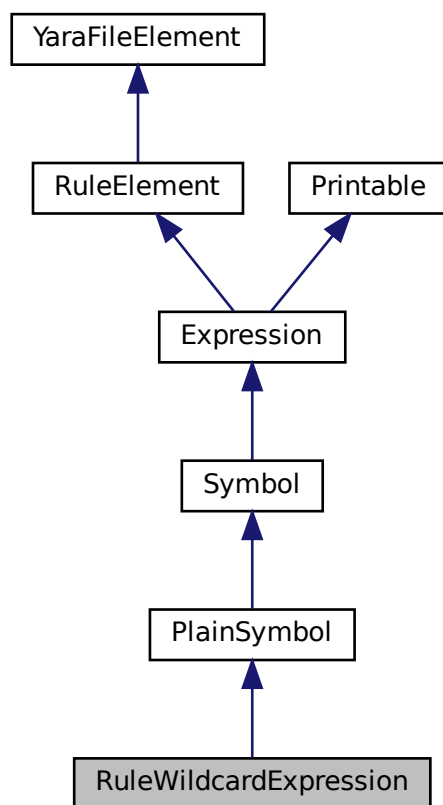
Implements [Printable](#).

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

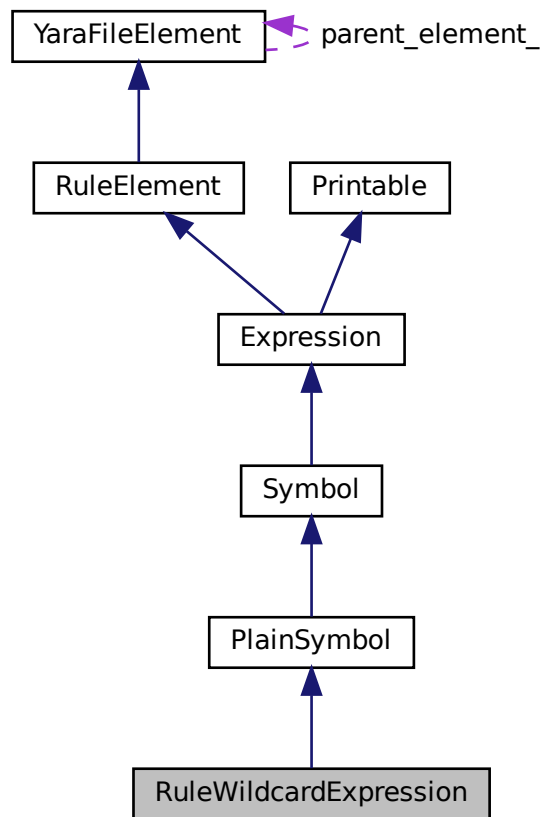
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.117 RuleWildcardExpression Class Reference

Inheritance diagram for RuleWildcardExpression:



Collaboration diagram for RuleWildcardExpression:



Public Member Functions

- **RuleWildcardExpression** (std::string_view string_id)
- void **accept** (Visitor *v) override

Additional Inherited Members

5.117.1 Member Function Documentation

5.117.1.1 accept()

```
void RuleWildcardExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Reimplemented from [PlainSymbol](#).

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

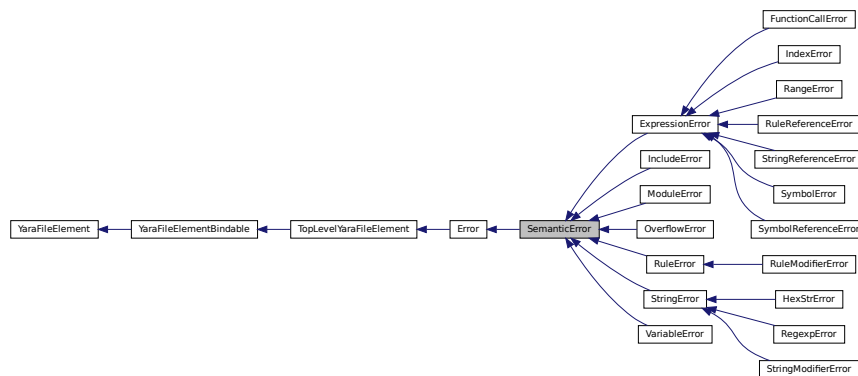
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.118 SemanticError Class Reference

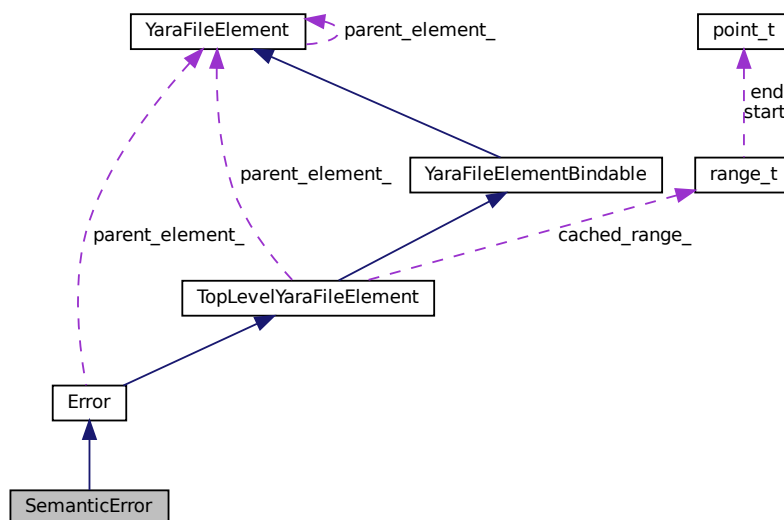
Base class for semantic errors.

```
#include <error.h>
```

Inheritance diagram for SemanticError:



Collaboration diagram for SemanticError:



Public Member Functions

- **SemanticError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [SemanticErrorException](#) exception ()
- bool [isGlobal](#) () const
- [SemanticError](#) * [setGlobal](#) (bool new_state)

Additional Inherited Members

5.118.1 Detailed Description

Base class for semantic errors.

5.118.2 Member Function Documentation

5.118.2.1 isGlobal()

```
bool SemanticError::isGlobal ( ) const
```

Returns global flag - this flag determines whether [SemanticError](#) was caused some token of [YaraFile](#) and so it cannot be fixed just by removing/updating this token (e.g. duplicated rule in included file -> we don't know which rule will user delete). Token with this token set to true are completely removed while reparsing and builded again (if there were not fixed).

5.118.2.2 setGlobal()

```
SemanticError * SemanticError::setGlobal (
    bool new_state )
```

Sets the state of global flag,

See also

[isGlobal](#)

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

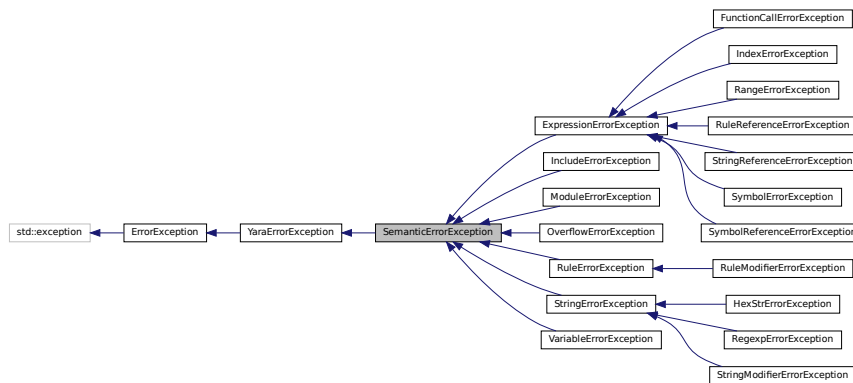
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/error.cpp](#)

5.119 SemanticErrorException Class Reference

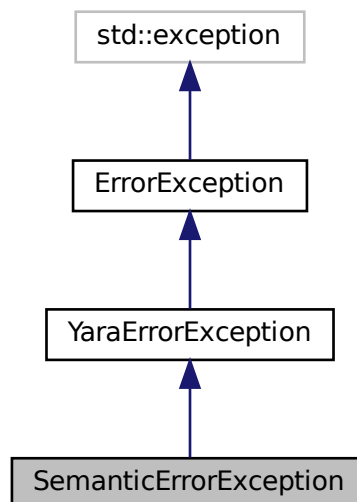
Exception representing semantic errors in parsed source code.

```
#include <error.h>
```

Inheritance diagram for SemanticErrorException:



Collaboration diagram for SemanticErrorException:



Public Member Functions

- **SemanticErrorException** (offset_t offset, size_t len, std::string msg={})

5.119.1 Detailed Description

Exception representing semantic errors in parsed source code.

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

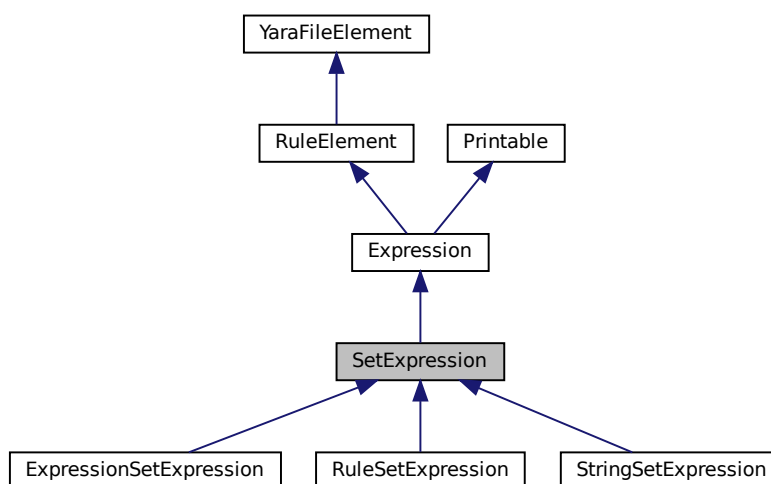
- `/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h`

5.120 SetExpression Class Reference

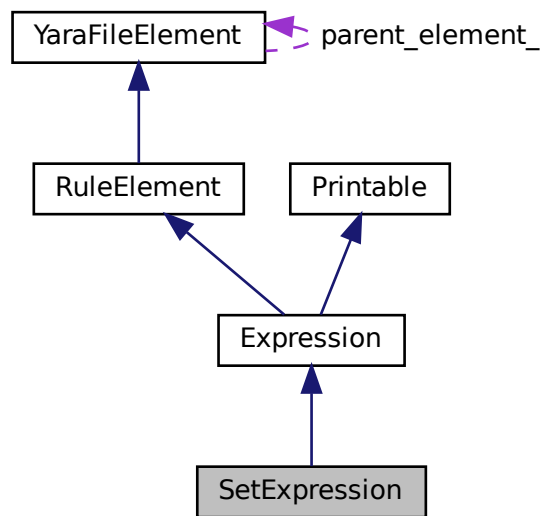
Base class for all set expressions.

```
#include <expression.h>
```

Inheritance diagram for SetExpression:



Collaboration diagram for SetExpression:



Public Member Functions

- **SetExpression** (std::vector< ExpressionPtr > elements)
- bool **isValid** (std::string_view &) const override
- bool **areOperandsValid** () const override
- bool **isComplete** () const override
- **Expression::Type** **getType** () const override
- const std::vector< ExpressionPtr > & **getElements** () const

Protected Attributes

- std::vector< ExpressionPtr > **elements_**

Additional Inherited Members

5.120.1 Detailed Description

Base class for all set expressions.

5.120.2 Member Function Documentation

5.120.2.1 areOperandsValid()

```
bool SetExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.120.2.2 getElements()

```
const std::vector< ExpressionPtr > & SetExpression::getElements ( ) const
```

Returns vector with all elements (pointers to expressions) of set

5.120.2.3 getType()

```
Expression::Type SetExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.120.2.4 isComplete()

```
bool SetExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.120.2.5 isValid()

```
bool SetExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

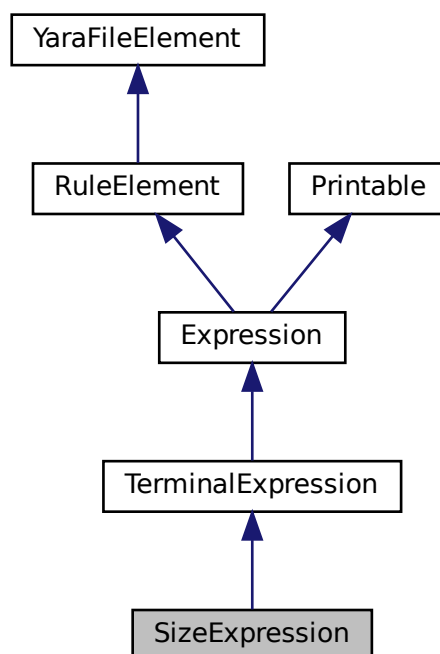
Implements [Expression](#).

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

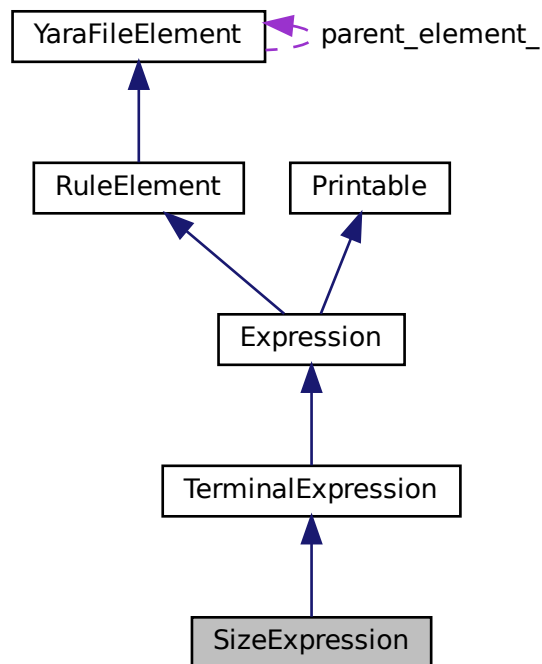
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.121 SizeExpression Class Reference

Inheritance diagram for SizeExpression:



Collaboration diagram for SizeExpression:



Public Types

- enum class **Unit** { **MB** , **KB** }

Public Member Functions

- **SizeExpression** (uint32_t value, SizeExpression::Unit unit)
- bool **isComplete** () const override
- **Expression::Type** **getType** () const override
- std::stringstream **getTextFormatted** () const override
- void **accept** (Visitor *v) override
- const SizeExpression::Unit & **getUnit** () const
- uint32_t **getValue** () const

Static Public Member Functions

- static std::string_view **unitToString** (SizeExpression::Unit unit)

Additional Inherited Members

5.121.1 Member Function Documentation

5.121.1.1 `accept()`

```
void SizeExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.121.1.2 `getTextFormatted()`

```
std::stringstream SizeExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.121.1.3 `getType()`

```
Expression::Type SizeExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.121.1.4 `isComplete()`

```
bool SizeExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.121.1.5 unitToString()

```
std::string_view SizeExpression::unitToString (
    SizeExpression::Unit unit ) [static]
```

Converts enum value to readable string with unit

Returns

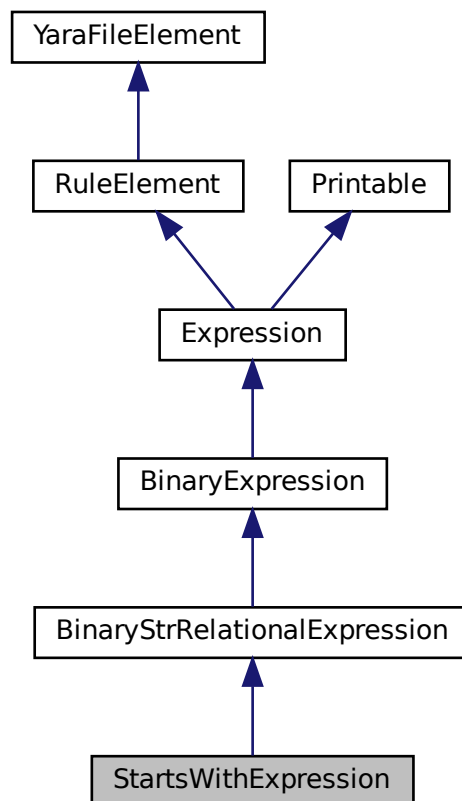
[String](#) view with readable representation of unit or with empty string

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

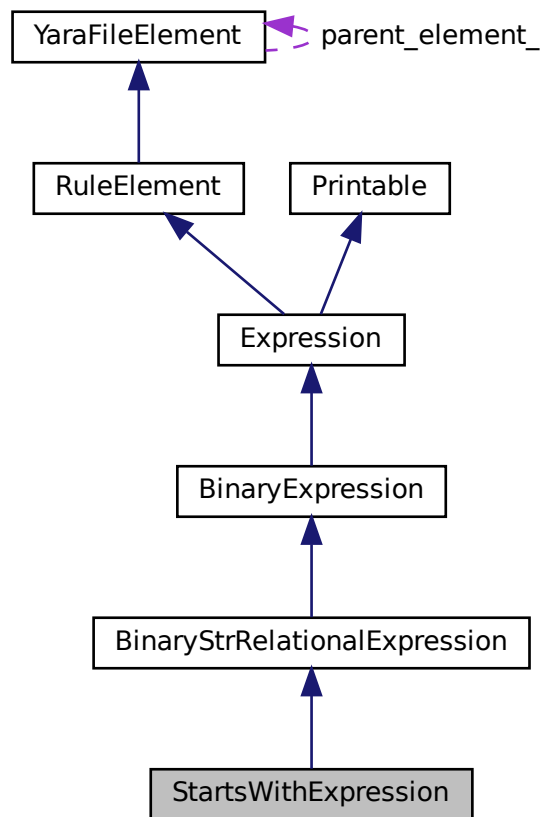
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.122 StartsWithExpression Class Reference

Inheritance diagram for StartsWithExpression:



Collaboration diagram for `StartsWithExpression`:



Public Member Functions

- **StartsWithExpression** (`ExpressionPtr lop`, `ExpressionPtr rop`)
- `std::string opsign ()` const override
- `void accept (Visitor *v)` override

Additional Inherited Members

5.122.1 Member Function Documentation

5.122.1.1 `accept()`

```
void StartsWithExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.122.1.2 opsign()

```
std::string StartsWithExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

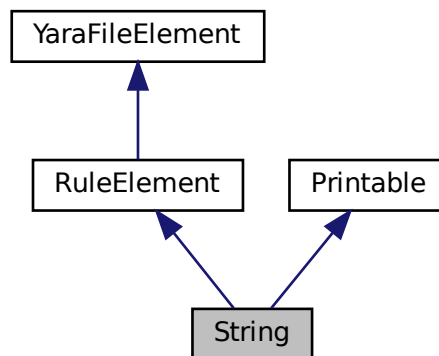
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.123 String Class Reference

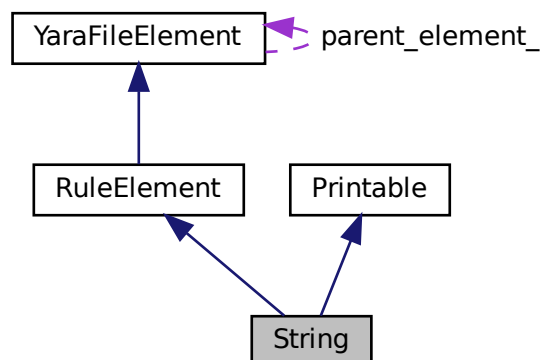
Represents yara string.

```
#include <string.h>
```

Inheritance diagram for String:



Collaboration diagram for String:



Public Types

- enum class [Type](#) { [Plain](#) , [Hex](#) , [Regexp](#) }
All valid types of yara strings.
- using **ValidModsTab** = std::unordered_map< [String::Type](#), std::unordered_set< [StringModifier::Type](#) > >

Public Member Functions

- String** ([String::Type](#) type)
- String** (const [String](#) &)=delete
- [String](#) & **operator=** (const [String](#) &)=delete
- void **setId** (const std::string &id)
- const std::string & **getId** (bool ignore_internal_id=false) const
- [String::Type](#) **getType** () const
- void **setContent** (const std::string &content)
- const std::string & **getContent** () const
- void **addModifier** (std::unique_ptr< [StringModifier](#) > &&new_mod)
- const [StringModifierContainer](#) & **getModifiers** () const
- bool **isAnonymous** () const
- void **setAnonymous** (bool is_anonymous)
- std::stringstream **getTextFormatted** () const override

Static Public Member Functions

- static std::string_view **typeToString** ([String::Type](#) type)

Additional Inherited Members

5.123.1 Detailed Description

Represents yara string.

5.123.2 Member Enumeration Documentation

5.123.2.1 Type

```
enum String::Type [strong]
```

All valid types of yara strings.

Enumerator

Plain	Plain text string.
Hex	Hex string.
Regexp	Regular expression string.

5.123.3 Member Function Documentation

5.123.3.1 addModifier()

```
void String::addModifier (
    std::unique_ptr< StringModifier > && new_mod )
```

Adds new modifier to modifier container < Check if there is already this type of modifier

< Check if modifier type is valid for current type of the string

5.123.3.2 getContent()

```
const std::string & String::getContent ( ) const
```

Returns reference to string content of the string

5.123.3.3 getId()

```
const std::string & String::getId (
    bool ignore_internal_id = false ) const
```

Returns reference to string with id of string

Parameters

<i>ignore_internal_id</i>	If it is set to true, id of anonymous string is ignored and empty string is returned (false by default)
---------------------------	---

5.123.3.4 getModifiers()

```
const StringModifierContainer & String::getModifiers ( ) const
```

Returns reference to container with string modifier

5.123.3.5 getTextFormatted()

```
std::stringstream String::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.123.3.6 `getType()`

```
String::Type String::getType ( ) const
```

Returns type of string

5.123.3.7 `isAnonymous()`

```
bool String::isAnonymous ( ) const
```

Returns value of anonymous flag of the string

Note

Anonymous flag determines, whether id of string was only defined as '\$'

5.123.3.8 `setAnonymous()`

```
void String::setAnonymous (
    bool is_anonymous )
```

Sets the value of anonymous flag of string

5.123.3.9 `setContent()`

```
void String::setContent (
    const std::string & content )
```

Sets the content of string

Note

Content is represented only as string

5.123.3.10 `setId()`

```
void String::setId (
    const std::string & id )
```

Sets id of string ('\$' must be omitted)

5.123.3.11 typeToString()

```
std::string_view String::typeToString (
    String::Type type ) [static]
```

Converts [String::Type](#) to readable string view

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[string.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[string.cpp](#)

5.124 string_hash_t Struct Reference

Hash struct for [SymTab](#), provides heterogeneous comparison lookup.

```
#include <types.h>
```

Public Types

- using **hash_type** = std::hash< std::string_view >
- using **is_transparent** = void

Public Member Functions

- std::size_t **operator()** (std::string_view sv) const
- std::size_t **operator()** (const std::string &str) const

5.124.1 Detailed Description

Hash struct for [SymTab](#), provides heterogeneous comparison lookup.

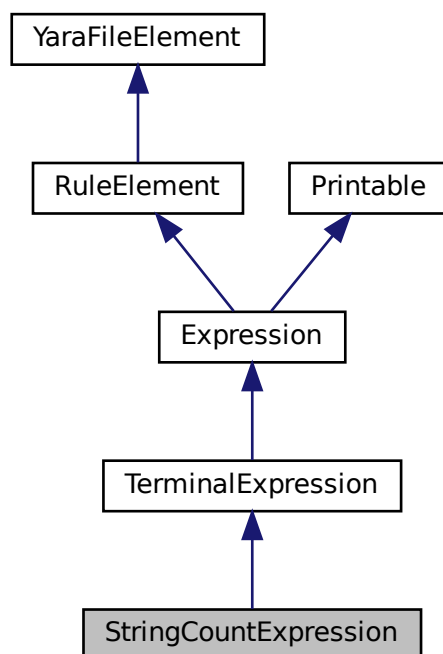
Based on: https://en.cppreference.com/w/cpp/container/unordered_map/find
Thanks to this, lookup in this [SymTab](#) can be done by std::string as well as with std::string_view (so, it is possible to avoid unnecessary allocation)

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

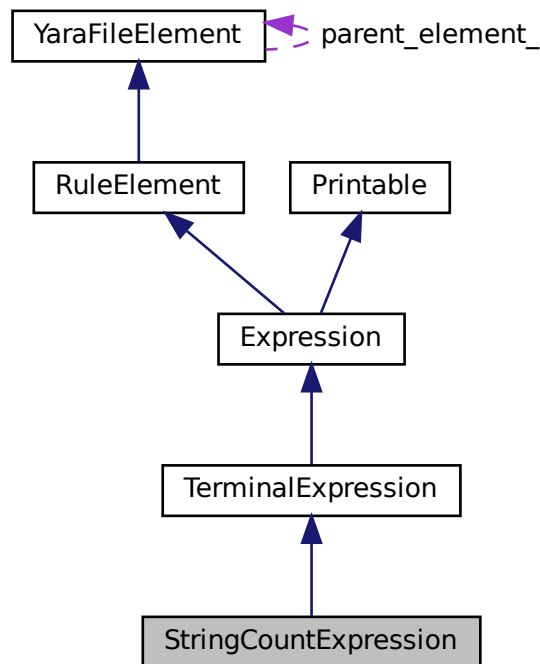
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[types.h](#)

5.125 StringCountExpression Class Reference

Inheritance diagram for StringCountExpression:



Collaboration diagram for StringCountExpression:



Public Member Functions

- **StringCountExpression** (std::string_view string_id)
- bool **isComplete** () const override
- **Expression::Type** **getType** () const override
- std::stringstream **getTextFormatted** () const override
- void **accept** (Visitor *v) override
- const std::string & **getId** () const

Additional Inherited Members

5.125.1 Member Function Documentation

5.125.1.1 accept()

```
void StringCountExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.125.1.2 `getTextFormatted()`

```
std::stringstream StringCountExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.125.1.3 `getType()`

```
Expression::Type StringCountExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.125.1.4 `isComplete()`

```
bool StringCountExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

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

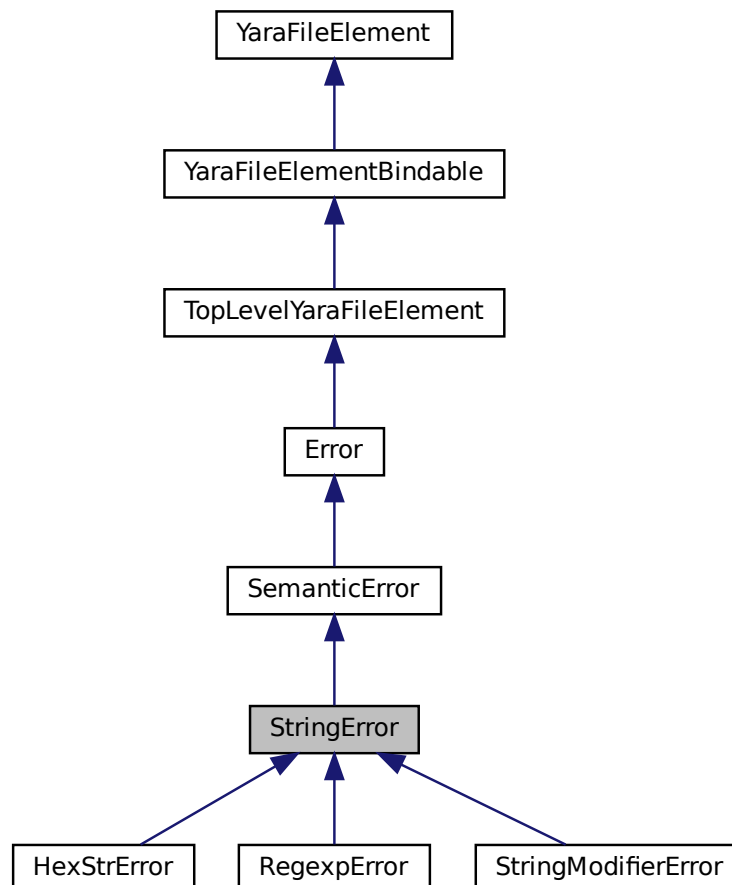
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.126 StringError Class Reference

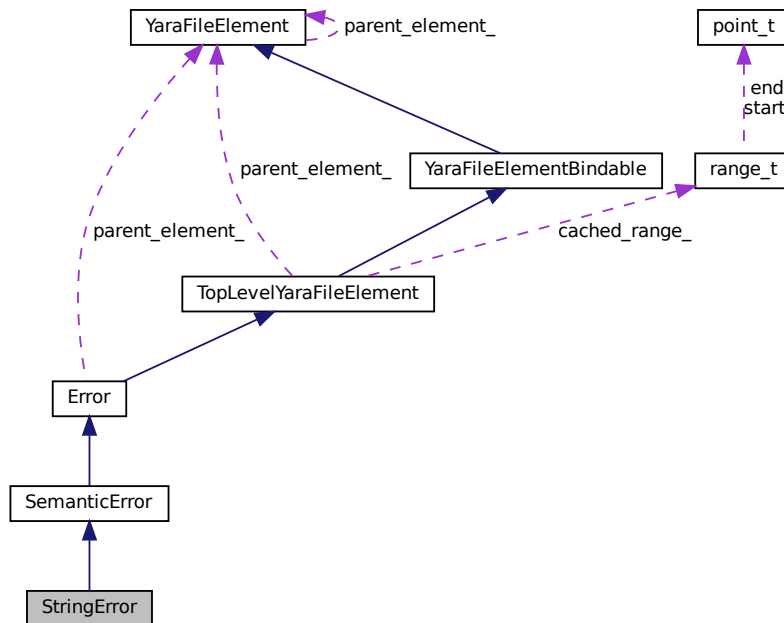
[Error](#) in string definition (e. g. collision of string identifiers)

```
#include <error.h>
```

Inheritance diagram for StringError:



Collaboration diagram for `StringError`:



Public Member Functions

- **`StringError`** (const std::string &desc, const offset_t &offset, const size_t &len)
- [`StringErrorException`](#) exception ()

Additional Inherited Members

5.126.1 Detailed Description

[Error](#) in string definition (e. g. collision of string identifiers)

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

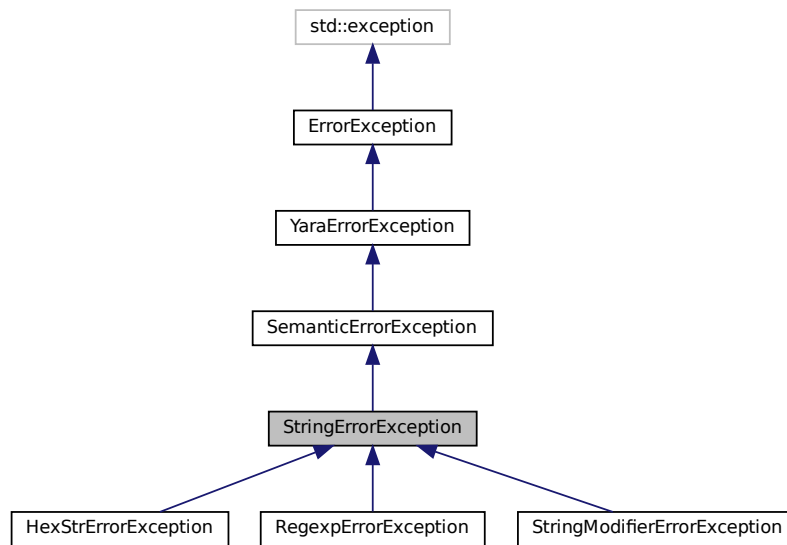
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.127 StringErrorException Class Reference

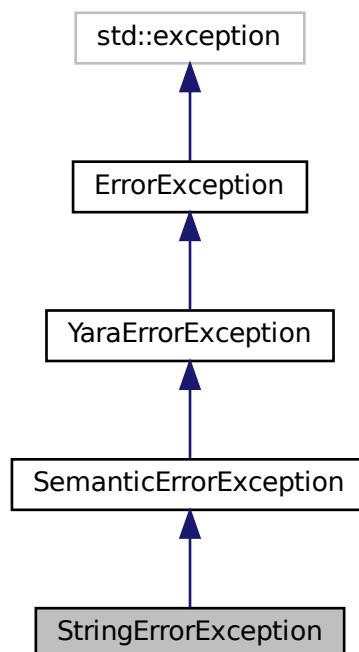
Exception for errors, that occurred in string definition (e. g. collision of string identifier)

```
#include <error.h>
```


Inheritance diagram for StringErrorException:



Collaboration diagram for StringErrorException:



Public Member Functions

- **StringErrorException** (offset_t offset, size_t len, std::string msg={})

5.127.1 Detailed Description

Exception for errors, that occurred in string definition (e. g. collision of string identifier)

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

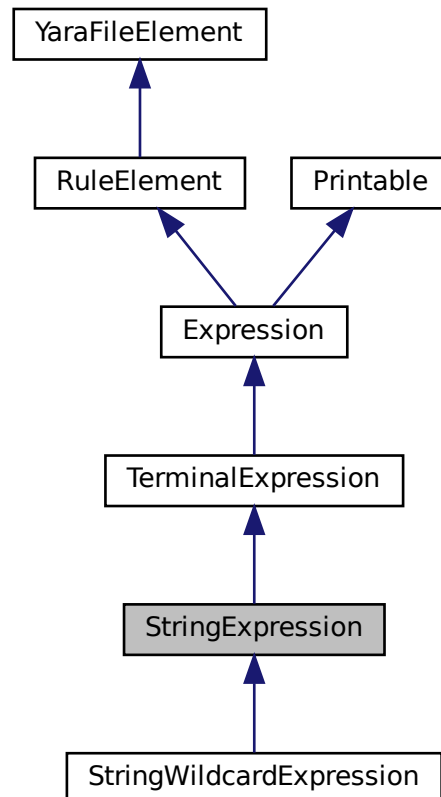
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.128 StringExpression Class Reference

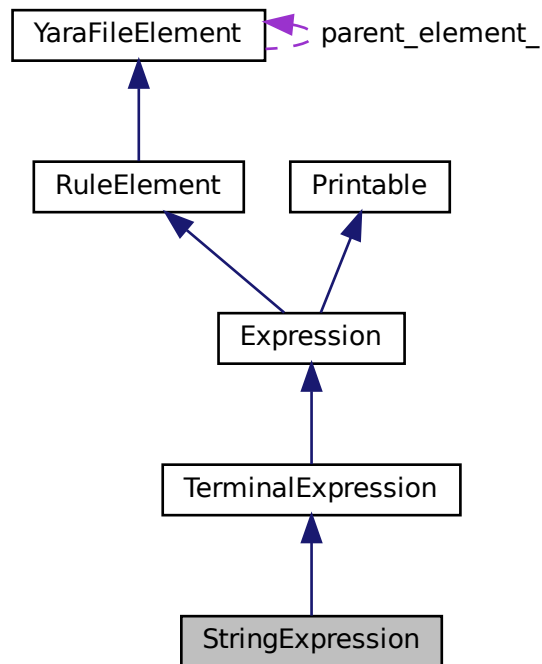
Represents string identifier in expression (\$<string_id>)

```
#include <expression.h>
```

Inheritance diagram for StringExpression:



Collaboration diagram for StringExpression:



Public Member Functions

- **StringExpression** (std::string_view string_id)
- bool **isComplete** () const override
- **Expression::Type** **getType** () const override
- std::stringstream **getTextFormatted** () const override
- void **accept** (Visitor *v) override
- const std::string & **getId** () const

Additional Inherited Members

5.128.1 Detailed Description

Represents string identifier in expression (\$<string_id>)

5.128.2 Member Function Documentation

5.128.2.1 accept()

```
void StringExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

Reimplemented in [StringWildcardExpression](#).

5.128.2.2 getTextFormatted()

```
std::stringstream StringExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.128.2.3 getType()

```
Expression::Type StringExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.128.2.4 isComplete()

```
bool StringExpression::isComplete ( ) const [override], [virtual]
```

Checks whether expression is complete - if it has all mandatory operands.

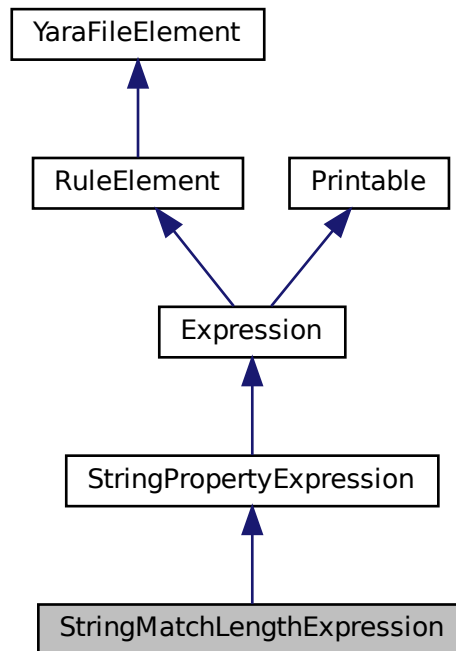
Implements [Expression](#).

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

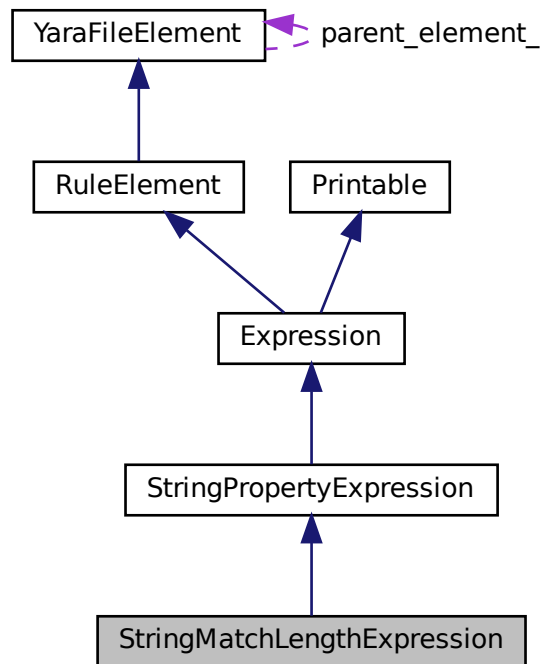
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.129 StringMatchLengthExpression Class Reference

Inheritance diagram for StringMatchLengthExpression:



Collaboration diagram for StringMatchLengthExpression:



Public Member Functions

- **StringMatchLengthExpression** (std::string_view string_id, ExpressionPtr index=nullptr)
- [Expression::Type](#) `getType` () const override
- std::stringstream `getTextFormatted` () const override
- void `accept` ([Visitor](#) *v) override

Additional Inherited Members

5.129.1 Member Function Documentation

5.129.1.1 `accept()`

```
void StringMatchLengthExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.129.1.2 getTextFormatted()

```
std::stringstream StringMatchLengthExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.129.1.3 getType()

```
Expression::Type StringMatchLengthExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

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

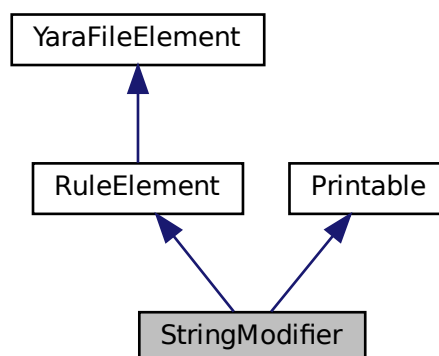
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.130 StringModifier Class Reference

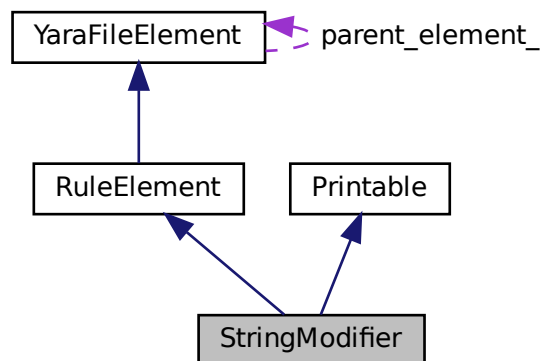
Represents modifier of yara string.

```
#include <string.h>
```

Inheritance diagram for StringModifier:



Collaboration diagram for StringModifier:



Public Types

- enum class [Type](#) {
NoCase , **Wide** , **Ascii** , **Xor** ,
Base64 , **Base64Wide** , **Fullword** , **Private** }
All valid types of yara strings modifiers.
- using **Range** = std::pair< uint8_t, uint8_t >
- using **Key** = uint8_t
- using **Alphabet** = std::array< char, [BASE64_ALPHA_LEN](#) >
- using **Arg** = std::variant< std::monostate, [Range](#), Alphabet, Key >

Public Member Functions

- **StringModifier** ([StringModifier::Type](#) type)
- [StringModifier::Type](#) **getType** () const
- void **setArg** (const Key &key)
- void **setArg** (const [Range](#) &range)
- void **setArg** (const Alphabet &alphabet)
- void **setArg** (const std::string &alphabet_str)
- bool **hasAnyArg** () const
- template<typename T >
 bool **hasArg** () const
- const Arg & **getArg** () const
- std::stringstream **getTextFormatted** () const override

Static Public Member Functions

- static [StringModifier::Type](#) **stringToType** (std::string_view string)
- static std::string_view **typeToString** ([StringModifier::Type](#) type)

Additional Inherited Members

5.130.1 Detailed Description

Represents modifier of yara string.

5.130.2 Member Function Documentation

5.130.2.1 getArg()

```
const StringModifier::Arg & StringModifier::getArg ( ) const
```

Returns reference to argument of modifier

5.130.2.2 getTextFormatted()

```
std::stringstream StringModifier::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.130.2.3 getType()

```
StringModifier::Type StringModifier::getType ( ) const
```

Returns type of string modifier

5.130.2.4 hasAnyArg()

```
bool StringModifier::hasAnyArg ( ) const
```

Checks if modifier has any argument

5.130.2.5 hasArg()

```
template<typename T >  
bool StringModifier::hasArg
```

Checks if modifier has argument of type T

5.130.2.6 setArg()

```
void StringModifier::setArg (
    const Key & key )
```

Sets the argument of modifier

5.130.2.7 stringToType()

```
StringModifier::Type StringModifier::stringToType (
    std::string_view string ) [static]
```

Converts string view to related [StringModifier::Type](#)

5.130.2.8 typeToString()

```
std::string_view StringModifier::typeToString (
    StringModifier::Type type ) [static]
```

Converts [StringModifier::Type](#) to readable string view

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

- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/string.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/string.cpp](#)

5.131 StringModifierContainer Class Reference

Iterable container for string modifier.

```
#include <string.h>
```

Classes

- class [iterator](#)
Iterator for [StringModifierContainer](#).

Public Member Functions

- **StringModifierContainer** (const [StringModifierContainer](#) &)=delete
- [StringModifierContainer](#) & **operator=** (const [StringModifierContainer](#) &)=delete
- [size_t](#) [size](#) () const
- bool [empty](#) () const
- bool [has](#) ([StringModifier::Type](#) type) const
- [StringModifier](#) * [get](#) ([StringModifier::Type](#) type) const
- [StringModifierContainer::iterator](#) [begin](#) () const
- [StringModifierContainer::iterator](#) [end](#) () const
- void [add](#) (std::unique_ptr< [StringModifier](#) > &&new_mod)
- void [erase](#) ([StringModifier::Type](#) type)

5.131.1 Detailed Description

Iterable container for string modifier.

5.131.2 Member Function Documentation

5.131.2.1 add()

```
void StringModifierContainer::add (
    std::unique_ptr< StringModifier > && new_mod )
```

Adds modifier to container

Note

If there is modifier with the same type it is erased

5.131.2.2 begin()

```
StringModifierContainer::iterator StringModifierContainer::begin ( ) const
```

Returns begin iterator of container

Returns

iterator of container

5.131.2.3 empty()

```
bool StringModifierContainer::empty ( ) const
```

Checks if there is at least one string modifier in container

5.131.2.4 end()

```
StringModifierContainer::iterator StringModifierContainer::end ( ) const
```

Returns end iterator of container

Returns

iterator of container

5.131.2.5 erase()

```
void StringModifierContainer::erase (
    StringModifier::Type type )
```

Erases the modifier with given type

Note

If there is no such element, nothing is done

5.131.2.6 get()

```
StringModifier * StringModifierContainer::get (
    StringModifier::Type type ) const
```

Returns pointer to string modifier in container with given type

Returns

ptr to string modifier

5.131.2.7 has()

```
bool StringModifierContainer::has (
    StringModifier::Type type ) const
```

Checks if there is string modifier with given type in container

5.131.2.8 size()

```
size_t StringModifierContainer::size ( ) const
```

Returns number of elements in string modifier container

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

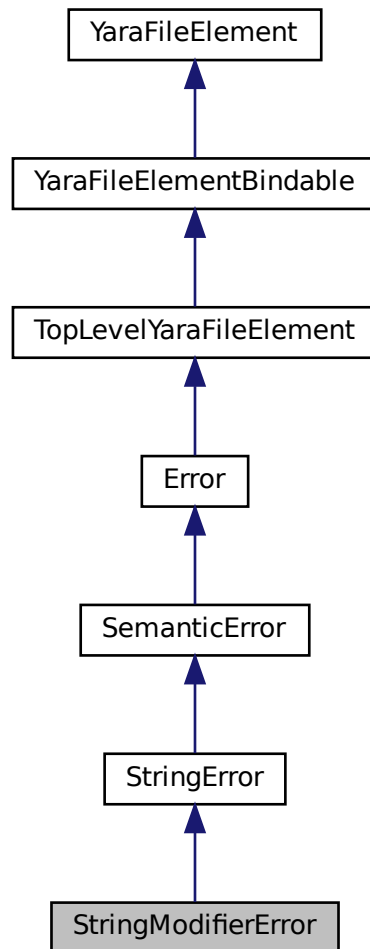
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[string.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[string.cpp](#)

5.132 StringModifierError Class Reference

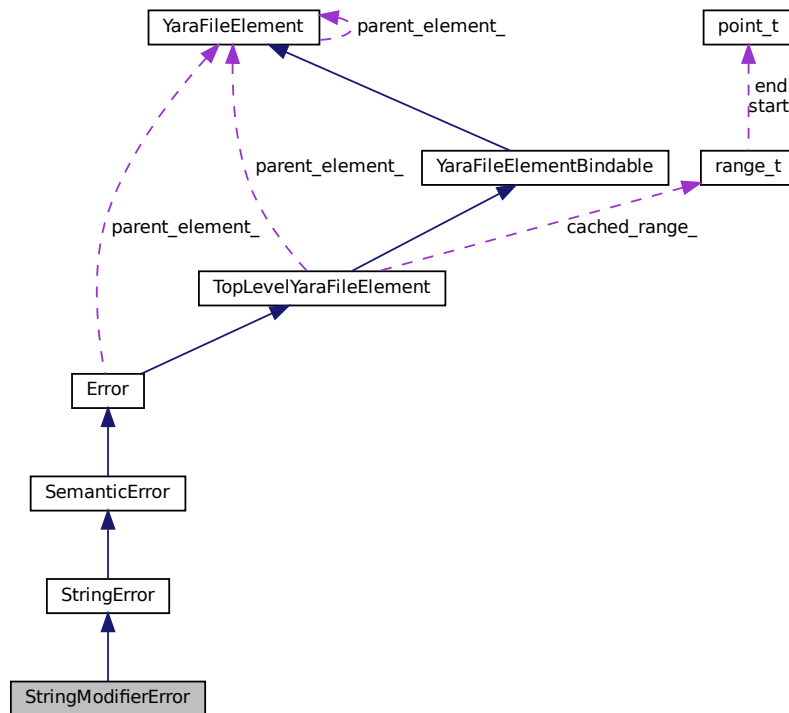
[Error](#) of string modifier (e. g. invalid type of string modifier for specific string)

```
#include <error.h>
```

Inheritance diagram for StringModifierError:



Collaboration diagram for StringModifierError:



Public Member Functions

- **StringModifierError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [StringModifierErrorException](#) exception ()

Additional Inherited Members

5.132.1 Detailed Description

[Error](#) of string modifier (e. g. invalid type of string modifier for specific string)

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

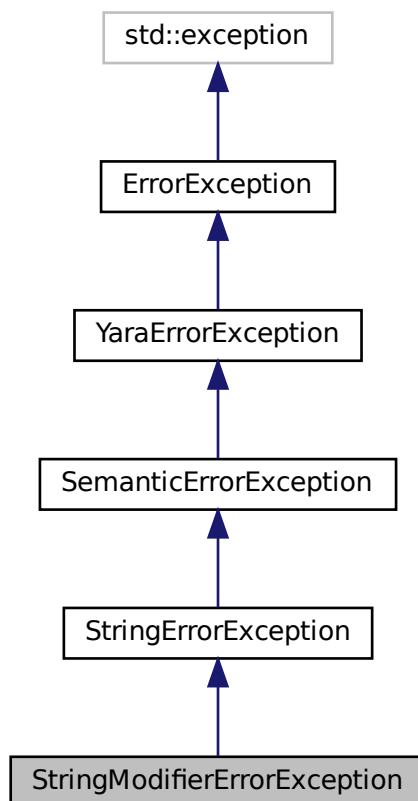
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.133 StringModifierErrorException Class Reference

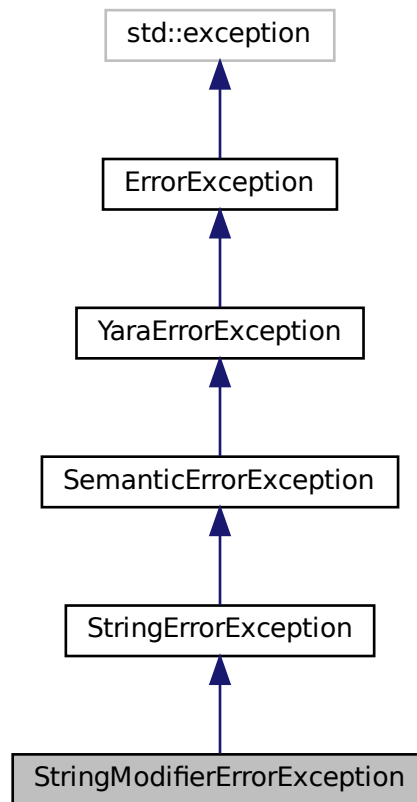
Exception error of string modifier (e. g, invalid combination of string type and modifier)

```
#include <error.h>
```

Inheritance diagram for StringModifierErrorException:



Collaboration diagram for StringModifierErrorException:



Public Member Functions

- **StringModifierErrorException** (offset_t offset, size_t len, std::string msg={})

5.133.1 Detailed Description

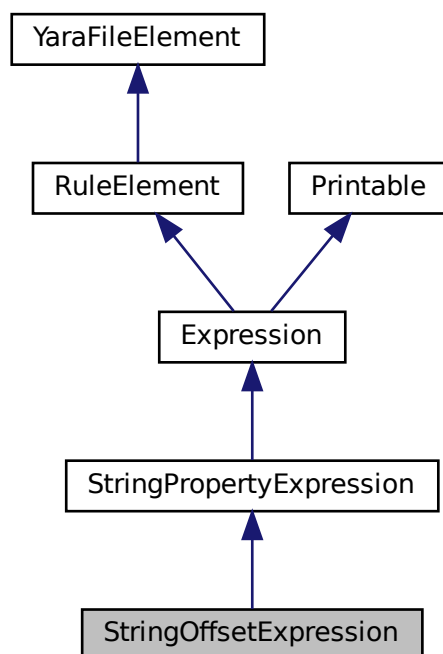
Exception error of string modifier (e. g, invalid combination of string type and modifier)

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

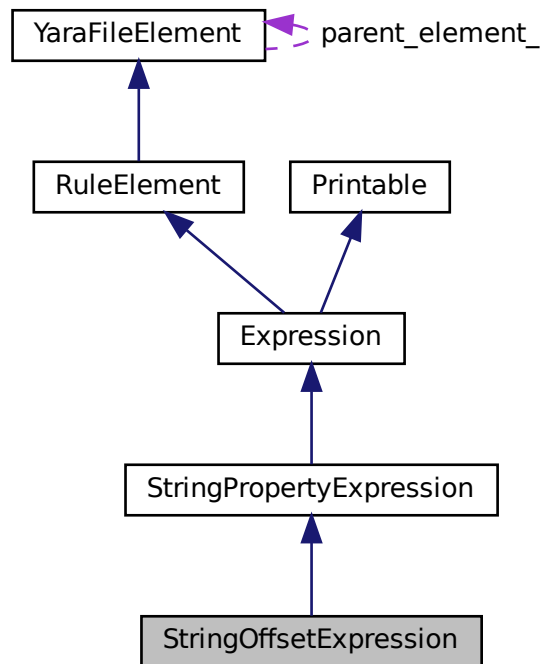
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.134 StringOffsetExpression Class Reference

Inheritance diagram for StringOffsetExpression:



Collaboration diagram for StringOffsetExpression:



Public Member Functions

- **StringOffsetExpression** (std::string_view string_id, ExpressionPtr index=nullptr)
- [Expression::Type](#) `getType` () const override
- std::stringstream `getTextFormatted` () const override
- void `accept` ([Visitor](#) *v) override

Additional Inherited Members

5.134.1 Member Function Documentation

5.134.1.1 `accept()`

```
void StringOffsetExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.134.1.2 getTextFormatted()

```
std::stringstream StringOffsetExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.134.1.3 getType()

```
Expression::Type StringOffsetExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

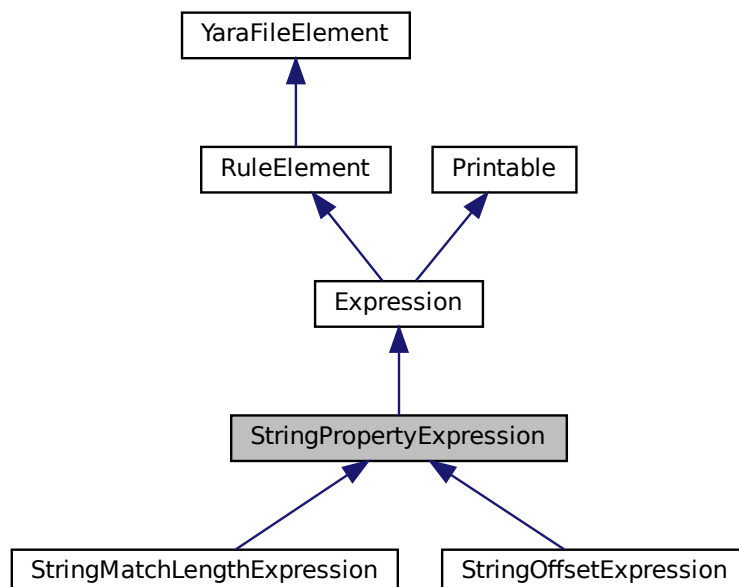
Implements [Expression](#).

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

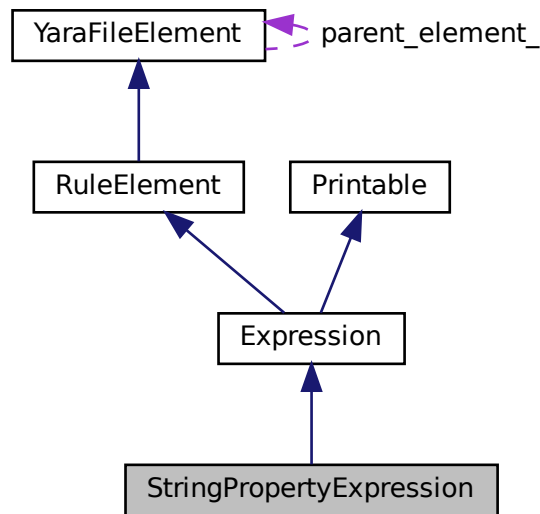
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.135 StringPropertyExpression Class Reference

Inheritance diagram for StringPropertyExpression:



Collaboration diagram for StringPropertyExpression:



Public Member Functions

- **StringPropertyExpression** (std::string_view string_id, ExpressionPtr index=nullptr)
- bool [areOperandsValid](#) () const override
- bool [isValid](#) (std::string_view &msg) const override
- bool [isComplete](#) () const override
- const std::string & **getId** () const
- const ExpressionPtr & **getIndex** () const

Protected Attributes

- std::string **string_id_** = {}
- ExpressionPtr **index_** = nullptr

Additional Inherited Members

5.135.1 Member Function Documentation

5.135.1.1 areOperandsValid()

```
bool StringPropertyExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.135.1.2 isComplete()

```
bool StringPropertyExpression::isComplete ( ) const [override], [virtual]
```

Checks whether expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.135.1.3 isValid()

```
bool StringPropertyExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

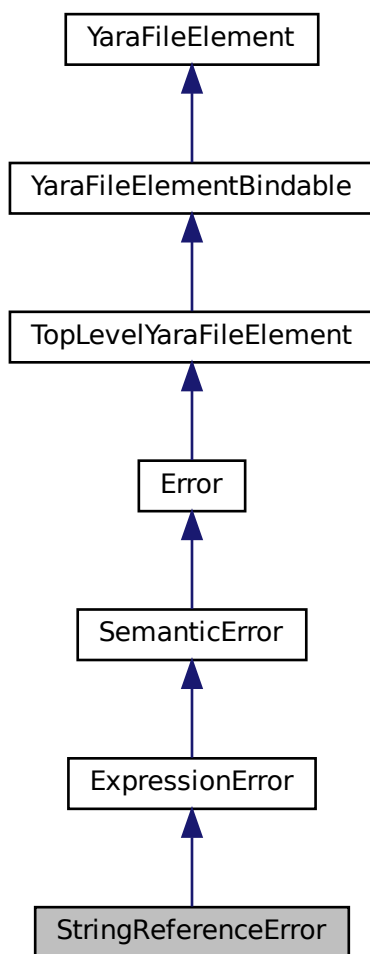
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.136 StringReferenceError Class Reference

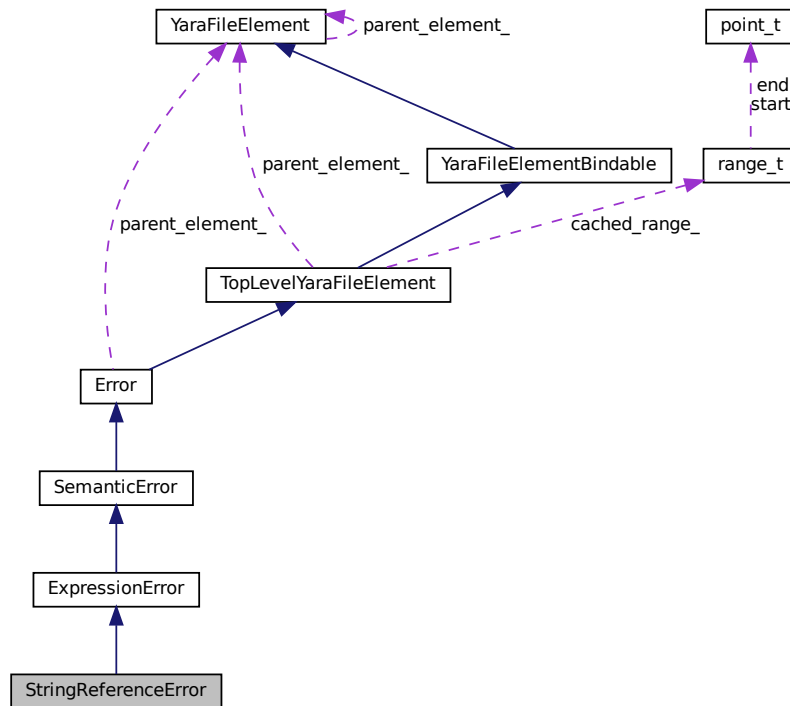
Bad string reference (referenced string does not exist)

```
#include <error.h>
```

Inheritance diagram for StringReferenceError:



Collaboration diagram for StringReferenceError:



Public Member Functions

- **StringReferenceError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [StringReferenceErrorException](#) **exception** ()

Additional Inherited Members

5.136.1 Detailed Description

Bad string reference (referenced string does not exists)

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

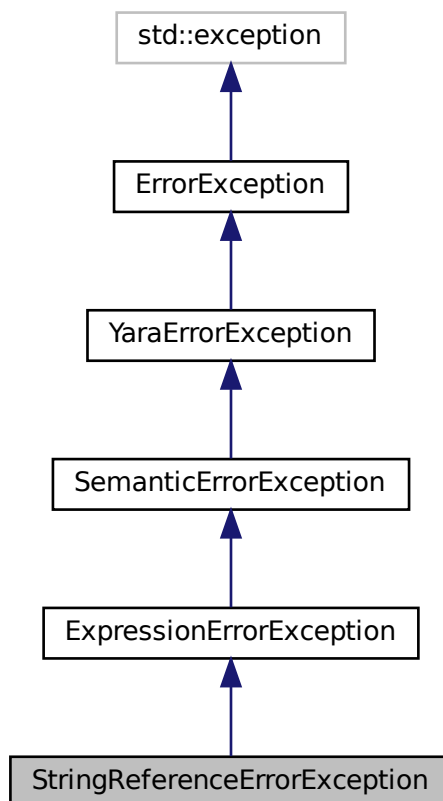
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.137 StringReferenceErrorException Class Reference

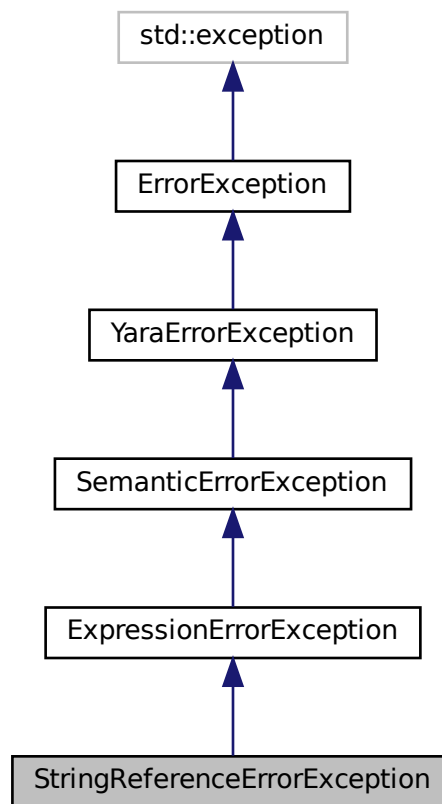
Exception for bad string reference (e. g. string does not exists)

```
#include <error.h>
```

Inheritance diagram for StringReferenceErrorException:



Collaboration diagram for StringReferenceErrorException:



Public Member Functions

- **StringReferenceErrorException** (offset_t offset, size_t len, std::string msg={})

5.137.1 Detailed Description

Exception for bad string reference (e. g. string does not exists)

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

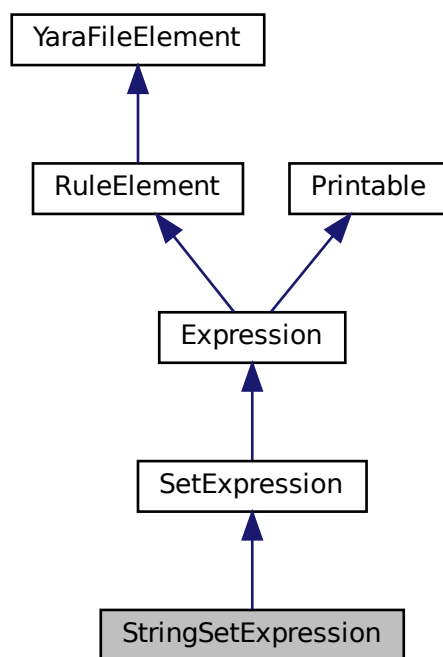
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.138 StringSetExpression Class Reference

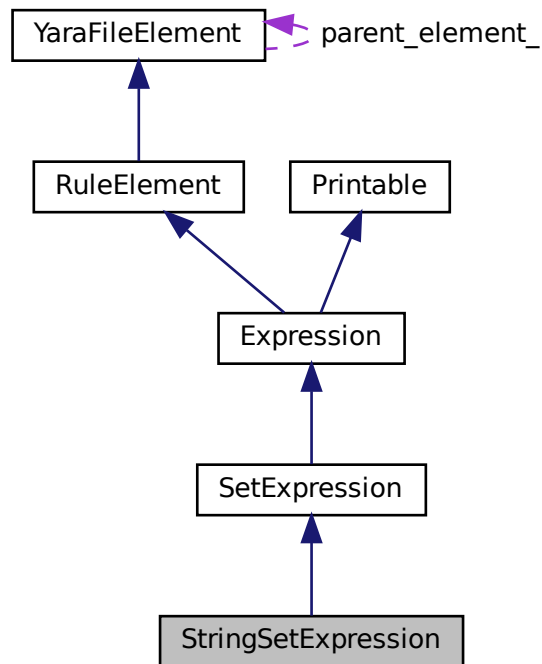
Represents set of strings.

```
#include <expression.h>
```

Inheritance diagram for StringSetExpression:



Collaboration diagram for StringSetExpression:



Public Member Functions

- **StringSetExpression** (std::vector< ExpressionPtr > elements)
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.138.1 Detailed Description

Represents set of strings.

5.138.2 Member Function Documentation

5.138.2.1 accept()

```
void StringSetExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.138.2.2 getTextFormatted()

```
std::stringstream StringSetExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

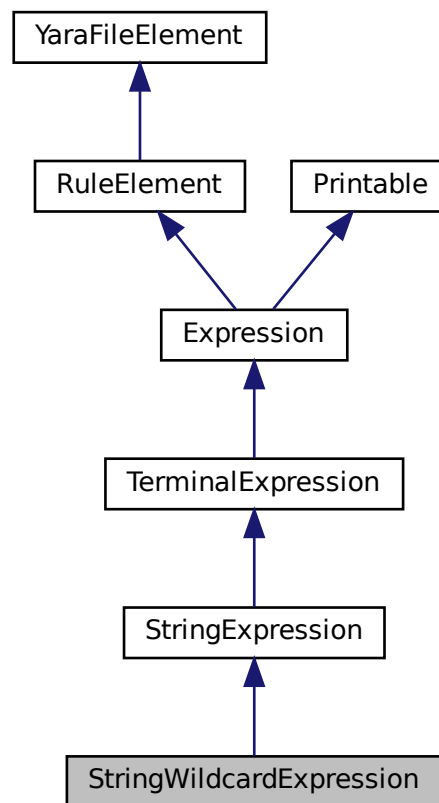
Implements [Printable](#).

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

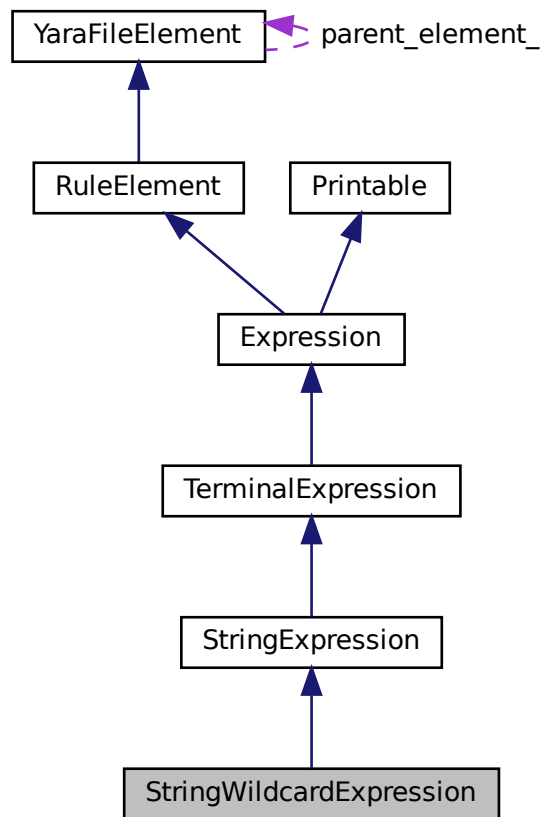
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.139 StringWildcardExpression Class Reference

Inheritance diagram for StringWildcardExpression:



Collaboration diagram for StringWildcardExpression:



Public Member Functions

- **StringWildcardExpression** (std::string_view string_id)
- void **accept** (Visitor *v) override

Additional Inherited Members

5.139.1 Member Function Documentation

5.139.1.1 accept()

```
void StringWildcardExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

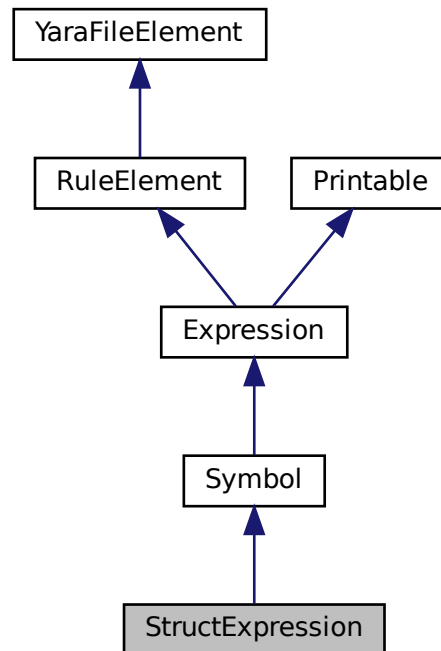
Reimplemented from [StringExpression](#).

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

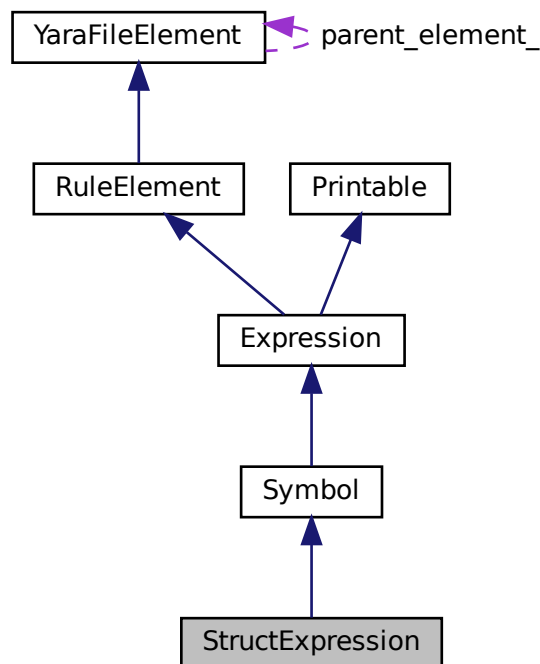
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.140 StructExpression Class Reference

Inheritance diagram for StructExpression:



Collaboration diagram for StructExpression:



Public Member Functions

- **StructExpression** (std::shared_ptr< [Symbol](#) > structure, std::shared_ptr< [Symbol](#) > member)
- bool [isComplete](#) () const override
- bool [areOperandsValid](#) () const override
- bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) ([Visitor](#) *v) override
- const std::shared_ptr< [Symbol](#) > & [getStructure](#) () const
- const std::shared_ptr< [Symbol](#) > & [getMember](#) () const

Additional Inherited Members

5.140.1 Member Function Documentation

5.140.1.1 accept()

```
void StructExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.140.1.2 areOperandsValid()

```
bool StructExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Reimplemented from [Symbol](#).

5.140.1.3 getTextFormatted()

```
std::stringstream StructExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Reimplemented from [Symbol](#).

5.140.1.4 getType()

```
Expression::Type StructExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Reimplemented from [Symbol](#).

5.140.1.5 isComplete()

```
bool StructExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Reimplemented from [Symbol](#).

5.140.1.6 isValid()

```
bool StructExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

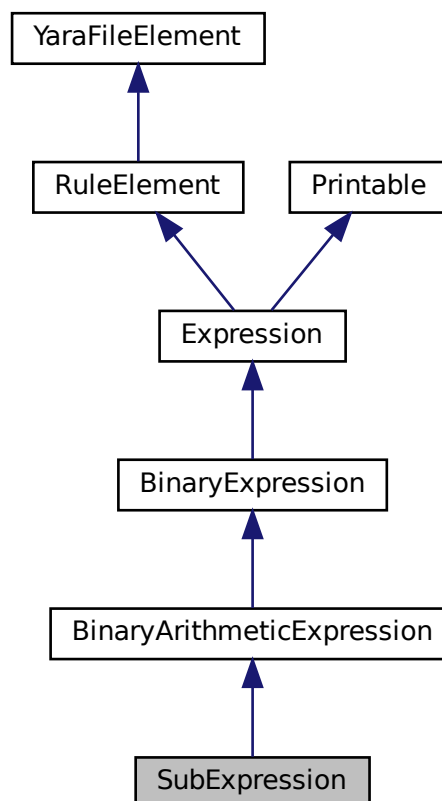
Reimplemented from [Symbol](#).

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

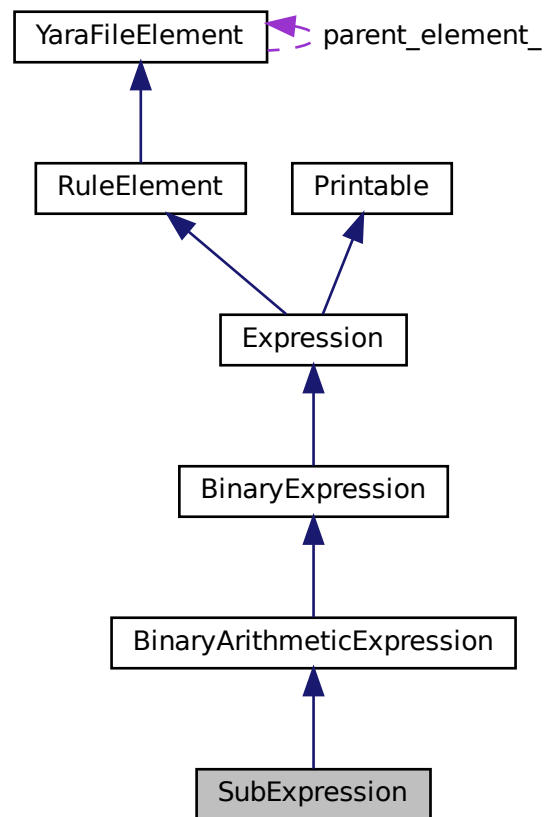
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.141 SubExpression Class Reference

Inheritance diagram for SubExpression:



Collaboration diagram for SubExpression:



Public Member Functions

- **SubExpression** (ExpressionPtr lop, ExpressionPtr rop)
- std::string [opsign](#) () const override
- void [accept](#) (Visitor *v) override

Additional Inherited Members

5.141.1 Member Function Documentation

5.141.1.1 [accept\(\)](#)

```
void SubExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.141.1.2 opsign()

```
std::string SubExpression::opsign ( ) const [inline], [override], [virtual]
```

Returns the string with printable sign or symbol of corresponding, that is represented by specialized class.

Implements [BinaryExpression](#).

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

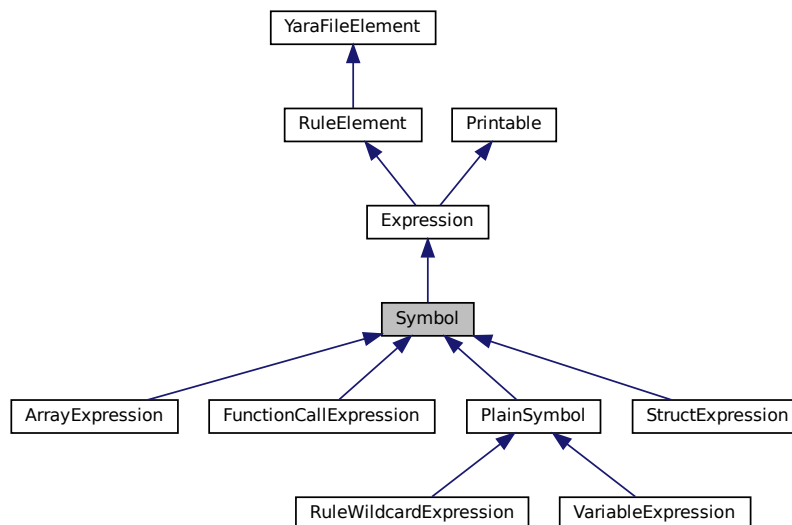
- </home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h>

5.142 Symbol Class Reference

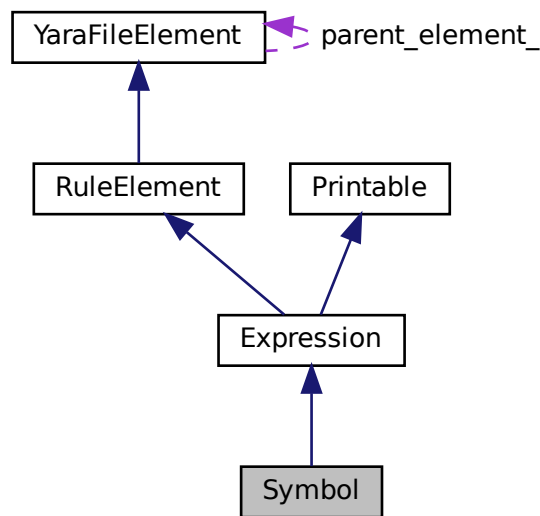
Base class for all symbols in expressions (structures, arrays, variables function calls...)

```
#include <expression.h>
```

Inheritance diagram for Symbol:



Collaboration diagram for Symbol:



Public Types

- enum class **Type** {
Unknown , **Function** , **Array** , **Dict** ,
Struct , **Value** }
Type of symbol.

Public Member Functions

- bool **isComplete** () const override
- bool **areOperandsValid** () const override
- bool **isValid** (std::string_view &msg) const override
- void **setType** (Expression::Type type)
- Expression::Type **getType** () const override
- void **setSymType** (Symbol::Type type)
- Symbol::Type **getSymType** ()
- void **setContext** (const json *attr_array)
- const json * **getContext** () const
- std::stringstream **getTextFormatted** () const override
- void **setDefined** (bool defined)
- bool **isDefined** () const
- bool **isFunction** () const
- bool **isArray** () const
- bool **isDict** () const
- bool **isStruct** () const
- bool **isValue** () const

Protected Attributes

- [Expression::Type dtype_](#) = Expression::Type::Undefined
- [Symbol::Type type_](#) = Symbol::Type::Unknown
- const json * [ctx_](#) = nullptr
- bool [defined_](#) = false

Additional Inherited Members

5.142.1 Detailed Description

Base class for all symbols in expressions (structures, arrays, variables function calls...)

5.142.2 Member Function Documentation

5.142.2.1 areOperandsValid()

```
bool Symbol::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

Reimplemented in [FunctionCallExpression](#), [StructExpression](#), and [ArrayExpression](#).

5.142.2.2 getContext()

```
const json * Symbol::getContext ( ) const
```

Gets pointer to JSON context of the symbol

5.142.2.3 getSymType()

```
Symbol::Type Symbol::getSymType ( )
```

Gets symbol type

5.142.2.4 getTextFormatted()

```
std::stringstream Symbol::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

Reimplemented in [PlainSymbol](#), [FunctionCallExpression](#), [StructExpression](#), and [ArrayExpression](#).

5.142.2.5 getType()

```
Expression::Type Symbol::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

Reimplemented in [FunctionCallExpression](#), and [StructExpression](#).

5.142.2.6 isComplete()

```
bool Symbol::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

Reimplemented in [FunctionCallExpression](#), [StructExpression](#), and [ArrayExpression](#).

5.142.2.7 isDefined()

```
bool Symbol::isDefined ( ) const
```

Gets the value of defined flag. It is used to determine, whether symbol is defined or not

5.142.2.8 isValid()

```
bool Symbol::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

Reimplemented in [FunctionCallExpression](#), [StructExpression](#), and [ArrayExpression](#).

5.142.2.9 `setContext()`

```
void Symbol::setContext (
    const json * attr_array )
```

Sets the JSON context of the symbol

5.142.2.10 `setDefined()`

```
void Symbol::setDefined (
    bool defined )
```

Sets the 'defined' flag

5.142.2.11 `setSymType()`

```
void Symbol::setSymType (
    Symbol::Type type )
```

Sets symbol type (

See also

[Symbol::Type](#))

5.142.3 Member Data Documentation

5.142.3.1 `ctx_`

```
const json* Symbol::ctx_ = nullptr [protected]
```

Holds JSON context (related JSON object) of the symbol

5.142.3.2 defined_

```
bool Symbol::defined_ = false [protected]
```

Defined flag - it should be set to true if symbol is defined otherwise it should be false

5.142.3.3 dtype_

```
Expression::Type Symbol::dtype_ = Expression::Type::Undefined [protected]
```

Holds data type ([Expression::Type](#)) of symbol

5.142.3.4 stype_

```
Symbol::Type Symbol::stype_ = Symbol::Type::Unknown [protected]
```

Holds type of the symbol

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

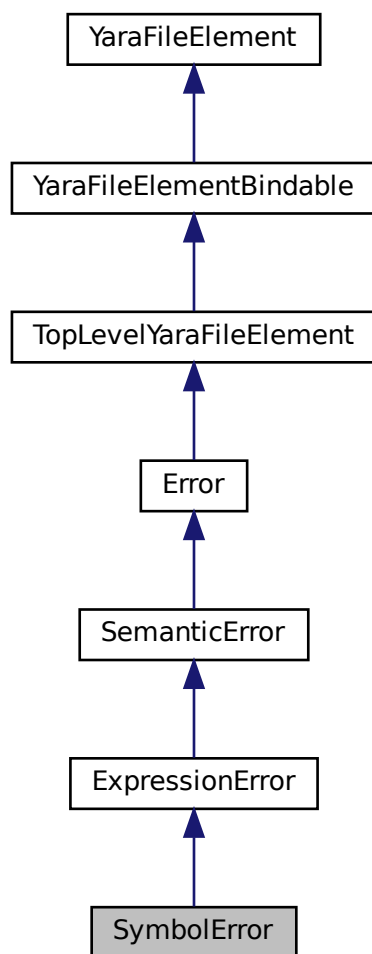
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.143 SymbolError Class Reference

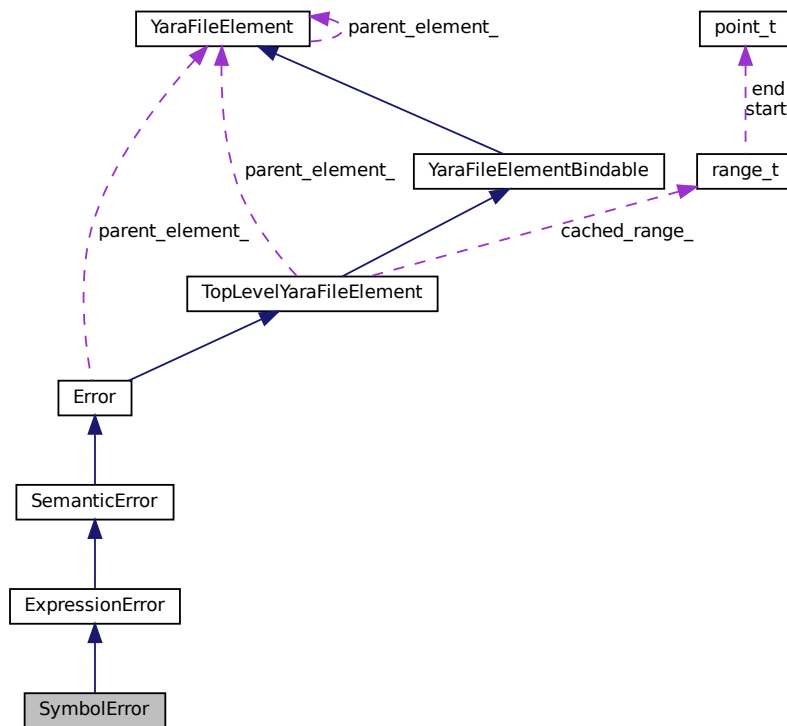
Bad usage of symbol.

```
#include <error.h>
```

Inheritance diagram for SymbolError:



Collaboration diagram for SymbolError:



Public Member Functions

- **SymbolError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [SymbolErrorException](#) exception ()

Additional Inherited Members

5.143.1 Detailed Description

Bad usage of symbol.

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

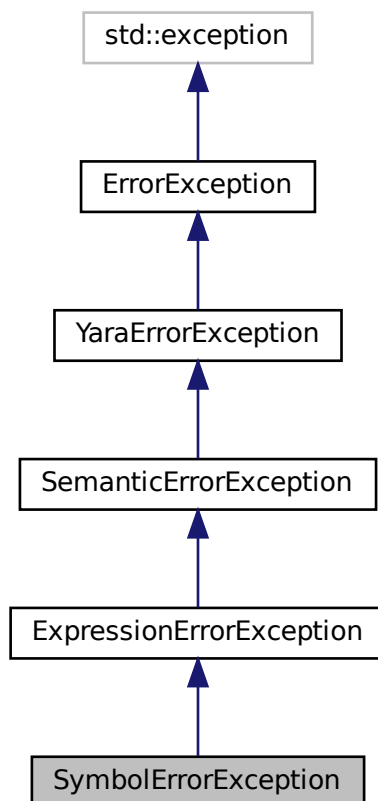
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.144 SymbolErrorException Class Reference

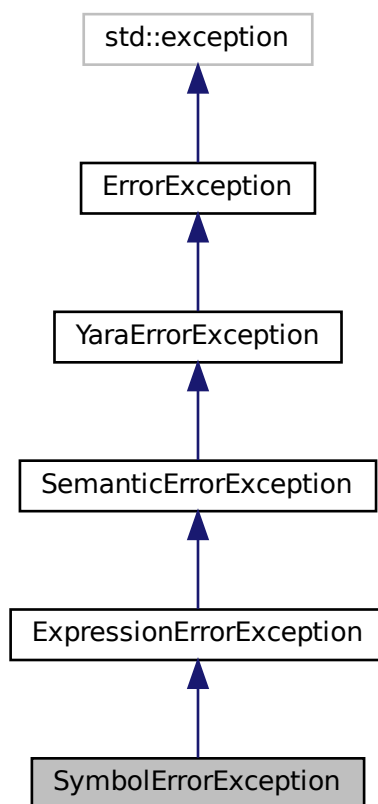
Exception for bad usage of symbol (e. g. symbol is used as structure, but it is value)

```
#include <error.h>
```

Inheritance diagram for SymbolErrorException:



Collaboration diagram for SymbolErrorException:



Public Member Functions

- **SymbolErrorException** (offset_t offset, size_t len, std::string msg={})

5.144.1 Detailed Description

Exception for bad usage of symbol (e. g. symbol is used as structure, but it is value)

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

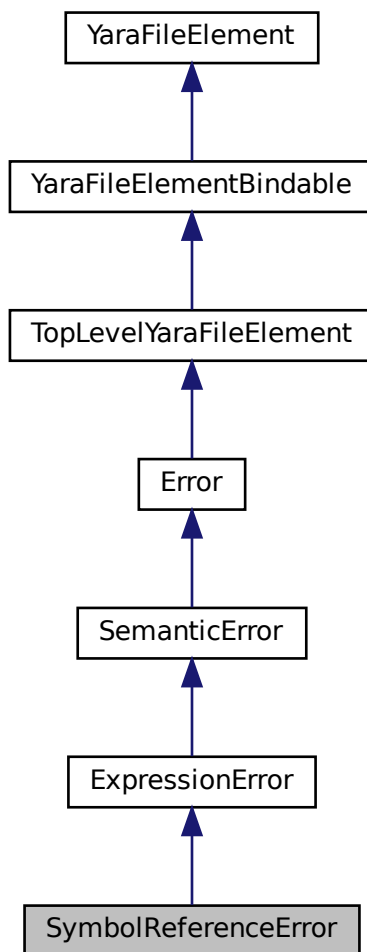
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.145 SymbolReferenceError Class Reference

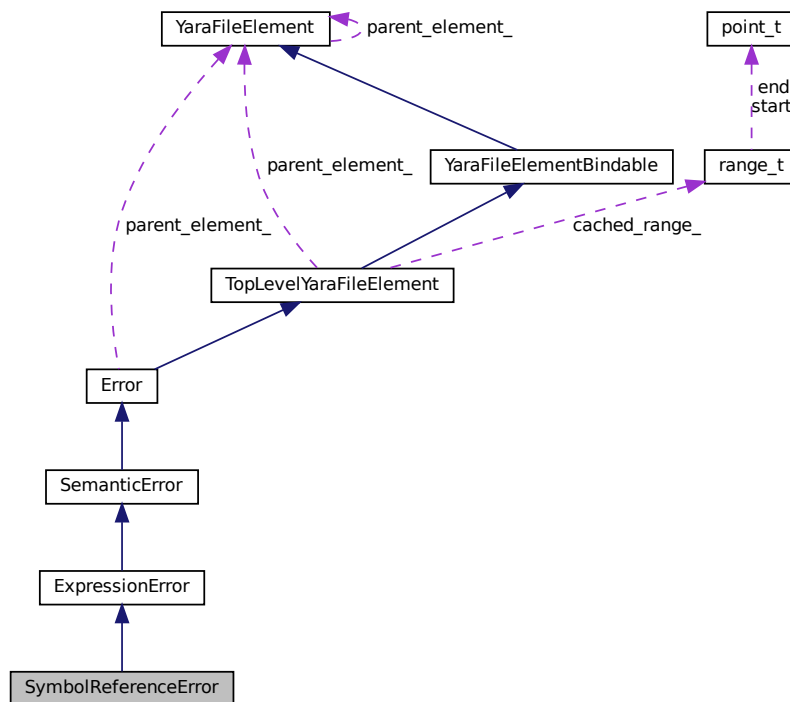
Bad symbol reference (e. g. reference to undefined symbol)

```
#include <error.h>
```

Inheritance diagram for SymbolReferenceError:



Collaboration diagram for SymbolReferenceError:



Public Member Functions

- **SymbolReferenceError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [SymbolReferenceErrorException](#) exception ()

Additional Inherited Members

5.145.1 Detailed Description

Bad symbol reference (e. g. reference to undefined symbol)

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

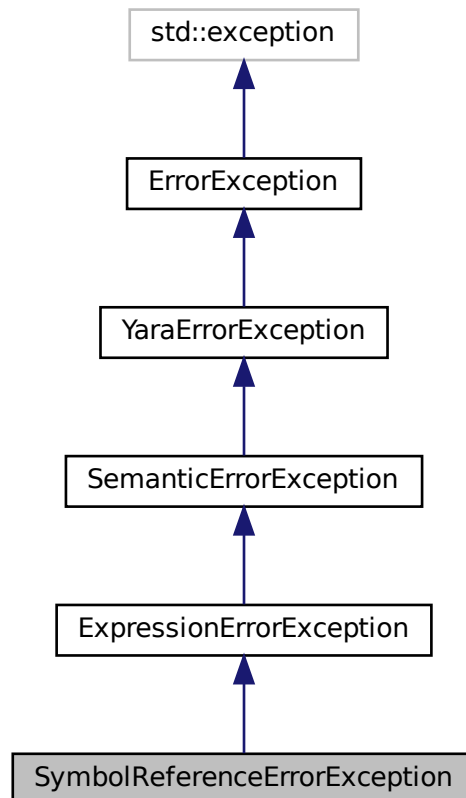
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.146 SymbolReferenceErrorException Class Reference

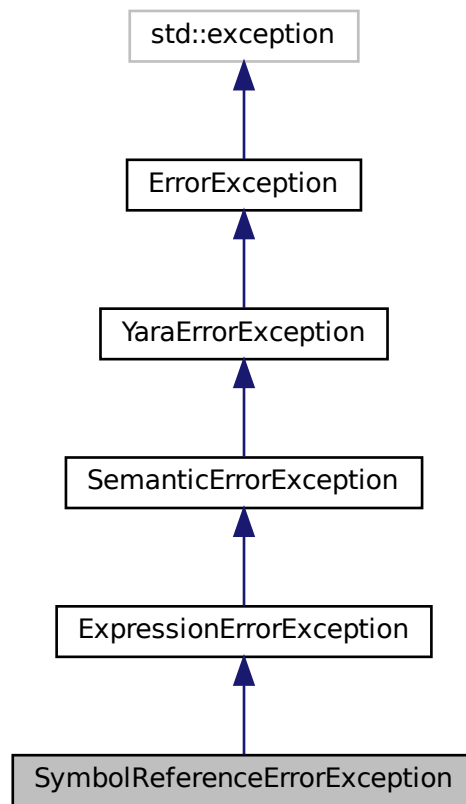
Exception for bad symbol reference (e. g. symbol does not exists)

```
#include <error.h>
```

Inheritance diagram for SymbolReferenceErrorException:



Collaboration diagram for SymbolReferenceErrorException:



Public Member Functions

- **SymbolReferenceErrorException** (offset_t offset, size_t len, std::string msg={})

5.146.1 Detailed Description

Exception for bad symbol reference (e. g. symbol does not exists)

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.147 SymTab< V > Class Template Reference

Tab for symbols with double indexing.

```
#include <symtab.h>
```

Classes

- class [iterator](#)
[SymTab](#) iterator (just a wrapper above `OffsetMap::const_iterator`)

Public Types

- using **OffsetMap** = `std::multimap< offset_t, V * >`
- using **IdMap** = `std::unordered_map< std::string, std::unique_ptr< V >, string_hash_t, std::equal_to<> >`
Holds the ownership of entries.

Public Member Functions

- **SymTab** (const [SymTab](#) &)=delete
- [SymTab](#) & **operator=** (const [SymTab](#) &)=delete
- [iterator](#) **begin** () const
- [iterator](#) **end** () const
- `size_t` **size** () const
- `bool` **empty** () const
- `const std::unique_ptr< V > &` **at** (const `std::string` &id) const
- `void` **insert** (`std::unique_ptr< V > &&`value)
- `const std::unique_ptr< V > &` **search** (const `std::string` &key) const
- `const std::unique_ptr< V > &` **search** (`std::string_view` key) const
- `bool` **exists** (const `std::string` &key) const
- `bool` **exists** (`std::string_view` key) const
- `void` **remove** (const `std::string` &id)
- `template<typename Fn >`
`uint32_t` **remove** (const `Fn` &f, `offset_t` start, `offset_t` end)
- `template<typename Fn >`
`uint32_t` **revRemoveUntil** (const `Fn` &f, `offset_t` rstart)
- `template<typename Fn >`
`void` **update** (const `Fn` &f, `offset_t` start, `offset_t` end)
- `void` **clear** ()
- `const IdMap &` **dataById** () const
- `const OffsetMap &` **dataByOffset** () const
- `V *` **operator[]** (`size_t` index) const
- `V *` **operator[]** (const `std::string` &id) const

5.147.1 Detailed Description

```
template<class V>
class SymTab< V >
```

Tab for symbols with double indexing.

Symbols are indexed by their unique id and by their offset.

5.147.2 Member Function Documentation

5.147.2.1 clear()

```
template<class V >
void SymTab< V >::clear
```

Clears the symbol tab

5.147.2.2 dataById()

```
template<class V >
const SymTab< V >::IdMap & SymTab< V >::dataById
```

Gets the data indexed by string id

5.147.2.3 dataByOffset()

```
template<class V >
const std::multimap< offset_t, V * > & SymTab< V >::dataByOffset
```

Gets the data indexed by offset

5.147.2.4 insert()

```
template<class V >
void SymTab< V >::insert (
    std::unique_ptr< V > && value )
```

Inserts the new entry to the symtab < Inserting entry to offset index

< Moving entry to id index (which owns the entry)

5.147.2.5 remove() [1/2]

```
template<class V >
template<typename Fn >
uint32_t SymTab< V >::remove (
    const Fn & f,
    offset_t start,
    offset_t end )
```

Removes elements with offset from start to end, that meet the condition specified by the lambda function (first arg)

Returns

Number of removed elements

< Check if condition is met

5.147.2.6 remove() [2/2]

```
template<class V >
void SymTab< V >::remove (
    const std::string & id )
```

Removes entry with given id < Pointers are the same

5.147.2.7 revRemoveUntil()

```
template<class V >
template<typename Fn >
uint32_t SymTab< V >::revRemoveUntil (
    const Fn & f,
    offset_t rstart )
```

Removes elements in reversed order until the given lambda function returns true

Returns

Number of removed elements

< Check if the condition is met

5.147.2.8 search()

```
template<class V >
const std::unique_ptr< V > & SymTab< V >::search (
    const std::string & key ) const
```

Returns pointer to entry with given key

5.147.2.9 update()

```
template<class V >
template<typename Fn >
void SymTab< V >::update (
    const Fn & f,
    offset_t start,
    offset_t end )
```

Updates the entries with offset from start to end by given lambda. Keys = offset and string id, can be also changed.
< Perform the update

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

- </home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/symtab.h>

5.148 SymTabDuplId< V > Class Template Reference

Almost same type of symbol table as the [SymTab](#), but in this case entries are allowed to have duplicated ids.

```
#include <symtab.h>
```

Classes

- class [iterator](#)
SymTabDuplId iterator.
- class [Pool](#)
Pool that owns all elements with the same id.

Public Types

- using **IdMap** = std::unordered_map< std::string, [Pool](#), [string_hash_t](#), std::equal_to<> >
- using **OffsetMap** = std::multimap< offset_t, V * >

Public Member Functions

- **SymTabDuplId** (const [SymTabDuplId](#) &)=delete
- **SymTabDuplId** & **operator=** (const [SymTabDuplId](#) &)=delete
- [iterator](#) **begin** () const
- [iterator](#) **end** () const
- size_t **size** () const
- bool **empty** () const
- const [SymTabDuplId](#)< V >::[Pool](#) & **at** (const std::string &id) const
- void **insert** (std::unique_ptr< V > &&new_entry)
- bool **exists** (const std::string &id) const
- bool **exists** (std::string_view id) const
- const [SymTabDuplId](#)< V >::[Pool](#) & **getAll** (const std::string &id) const
- const [SymTabDuplId](#)< V >::[Pool](#) & **getAll** (std::string_view id) const
- V * **get** (const std::string &id) const
- V * **get** (std::string_view id) const
- void **clear** ()
- void **remove** (const std::string &id)
- template<typename Fn >
uint32_t **remove** (const Fn &f, offset_t start, offset_t end)
- template<typename Fn >
uint32_t **revRemoveUntil** (const Fn &f, offset_t rstart)
- template<typename Fn >
void **update** (const Fn &f, offset_t start, offset_t end)
- const IdMap & **dataById** () const
- const OffsetMap & **dataByOffset** () const
- V * **operator[]** (size_t index) const
- const [SymTabDuplId](#)< V >::[Pool](#) & **operator[]** (const std::string &id) const

5.148.1 Detailed Description

```
template<class V>
class SymTabDuplId< V >
```

Almost same type of symbol table as the [SymTab](#), but in this case entries are allowed to have duplicated ids.

5.148.2 Member Function Documentation

5.148.2.1 at()

```
template<class V >
const SymTabDuplId< V >::Pool & SymTabDuplId< V >::at (
    const std::string & id ) const
```

Returns the vector of values stored under given string id

5.148.2.2 clear()

```
template<class V >
void SymTabDuplId< V >::clear
```

Clears the symbol table

5.148.2.3 exists()

```
template<class V >
bool SymTabDuplId< V >::exists (
    const std::string & id ) const
```

Check whether entry with given id exists or not.

5.148.2.4 getAll()

```
template<class V >
const SymTabDuplId< V >::Pool & SymTabDuplId< V >::getAll (
    const std::string & id ) const
```

Gets the vector of values stored under given string id

5.148.2.5 insert()

```
template<class V >
void SymTabDuplId< V >::insert (
    std::unique_ptr< V > && new_entry )
```

Inserts the new entry to the tab.

5.148.2.6 remove() [1/2]

```
template<class V >
template<typename Fn >
uint32_t SymTabDuplId< V >::remove (
    const Fn & f,
    offset_t start,
    offset_t end )
```

Removes all elements that meets the condition given as the first parameter (lambda returning bool) < Check if condition given by user is met

< Erase the pool

5.148.2.7 remove() [2/2]

```
template<class V >
void SymTabDuplId< V >::remove (
    const std::string & id )
```

Removes all entries with given id

5.148.2.8 revRemoveUntil()

```
template<class V >
template<typename Fn >
uint32_t SymTabDuplId< V >::revRemoveUntil (
    const Fn & f,
    offset_t rstart )
```

< Check if condition given by user is met

< Erase the empty pool

5.148.2.9 update()

```
template<class V >
template<typename Fn >
void SymTabDuplId< V >::update (
    const Fn & f,
    offset_t start,
    offset_t end )
```

Works similar as Symtab::update

Warning

Offset can be changed, but id of element cannot!

< Update the key

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

- </home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/symtab.h>

5.149 SymTabOffset< V > Class Template Reference

[Symbol](#) tab for symbols that are indexed only by offsets (which can be duplicated)

```
#include <symtab.h>
```

Classes

- class [iterator](#)
SymTabOffset iterator.

Public Types

- using **OffsetMap** = std::multimap< offset_t, std::unique_ptr< V > >

Public Member Functions

- **SymTabOffset** (const [SymTabOffset](#) &)=delete
- [SymTabOffset](#) & **operator=** (const [SymTabOffset](#) &)=delete
- [iterator](#) **begin** () const
- [iterator](#) **end** () const
- size_t **size** () const
- bool **empty** () const
- void **insert** (std::unique_ptr< V > &&new_entry)
- bool **exists** (const offset_t &offset) const
- const std::unique_ptr< V > & **get** (const offset_t &offset) const
- void **clear** ()
- void **remove** (const offset_t &offset)
- void **remove** (const [SymTabOffset](#)< V >::OffsetMap::const_iterator &it)
- template<typename Fn >
uint32_t **remove** (const Fn &f)
- template<typename Fn >
uint32_t **remove** (const Fn &f, offset_t start, offset_t end)
- template<typename Fn >
uint32_t **revRemoveUntil** (const Fn &f, offset_t rstart)
- template<typename Fn >
void **update** (const Fn &f)
- template<typename Fn >
void **update** (const Fn &f, offset_t start, offset_t end)
- const OffsetMap & **data** () const
- V * **operator[]** (size_t index) const

5.149.1 Detailed Description

```
template<class V>
class SymTabOffset< V >
```

[Symbol](#) tab for symbols that are indexed only by offsets (which can be duplicated)

5.149.2 Member Function Documentation

5.149.2.1 exists()

```
template<class V >
bool SymTabOffset< V >::exists (
    const offset_t & offset ) const
```

Checks if there is any entry on given offset

5.149.2.2 insert()

```
template<class V >
void SymTabOffset< V >::insert (
    std::unique_ptr< V > && new_entry )
```

Inserts the new entry

5.149.2.3 remove()

```
template<class V >
template<typename Fn >
uint32_t SymTabOffset< V >::remove (
    const Fn & f,
    offset_t start,
    offset_t end )
```

Removes the elements, that meet the condition given by lambda (first argument) between start and end offset. < Erase the elements that meet given condition

5.149.2.4 revRemoveUntil()

```
template<class V >
template<typename Fn >
uint32_t SymTabOffset< V >::revRemoveUntil (
    const Fn & f,
    offset_t rstart )
```

< Erase the elements that meet given condition

5.149.2.5 update() [1/2]

```
template<class V >
template<typename Fn >
void SymTabOffset< V >::update (
    const Fn & f )
```

Update all entries by given lambda function.

5.149.2.6 update() [2/2]

```
template<class V >
template<typename Fn >
void SymTabOffset< V >::update (
    const Fn & f,
    offset_t start,
    offset_t end )
```

Update all entries between start and end offsets. < Update the key

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

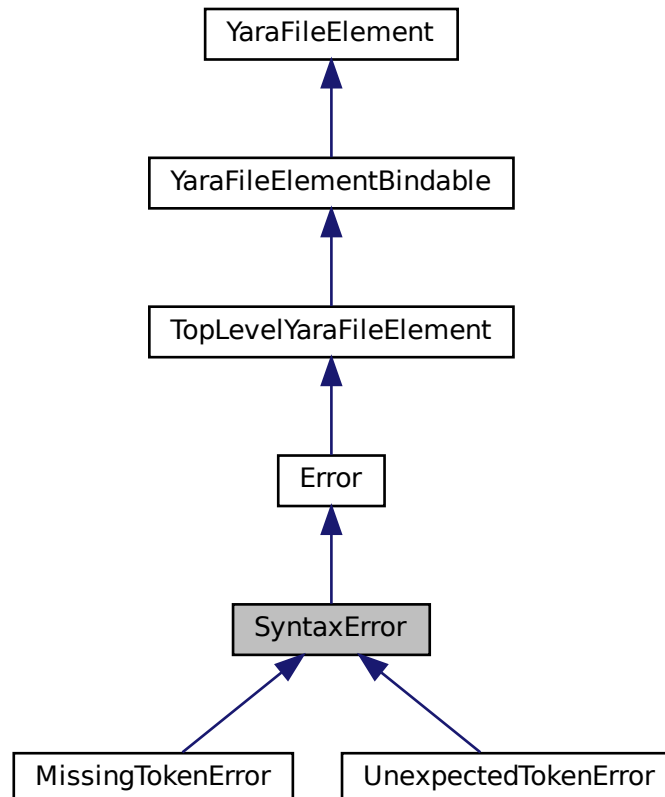
- </home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/symtab.h>

5.150 SyntaxError Class Reference

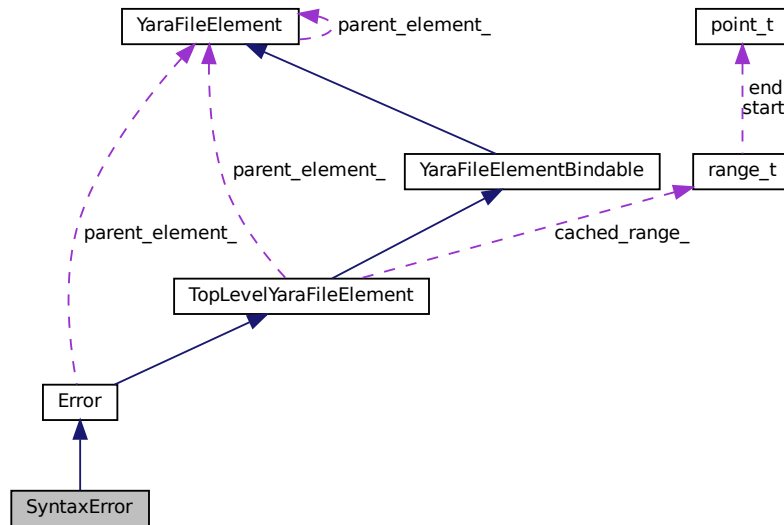
Base class for syntax errors.

```
#include <error.h>
```

Inheritance diagram for SyntaxError:



Collaboration diagram for SyntaxError:



Public Member Functions

- **SyntaxError** (const std::string &description, const offset_t &offset, const size_t &len)
- [SyntaxErrorException](#) exception ()

Additional Inherited Members

5.150.1 Detailed Description

Base class for syntax errors.

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

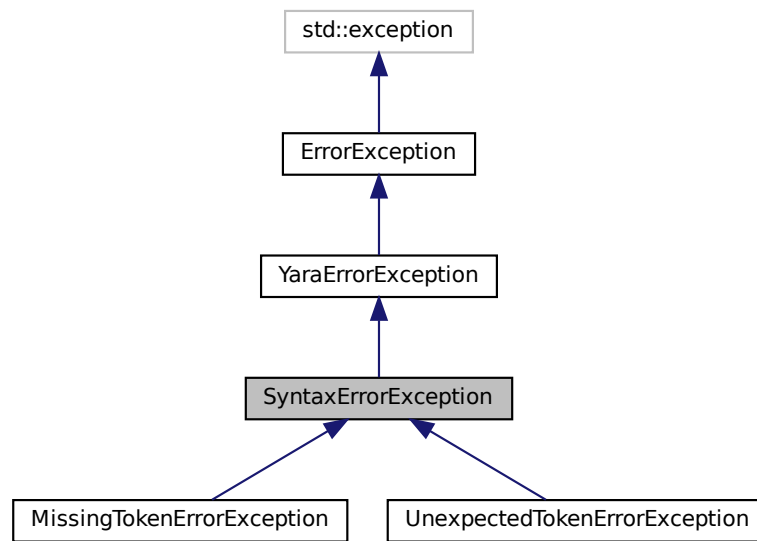
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[error.cpp](#)

5.151 SyntaxErrorException Class Reference

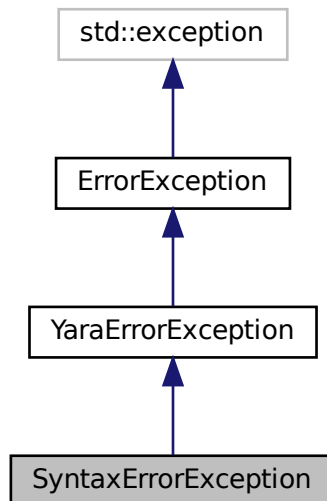
Exception representing syntax error in parsed source code.

```
#include <error.h>
```

Inheritance diagram for `SyntaxErrorException`:



Collaboration diagram for `SyntaxErrorException`:



Public Member Functions

- **`SyntaxErrorException`** (`offset_t` offset, `size_t` len, `std::string` msg={})

5.151.1 Detailed Description

Exception representing syntax error in parsed source code.

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

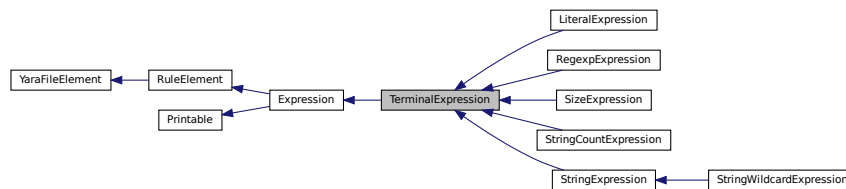
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h](#)

5.152 TerminalExpression Class Reference

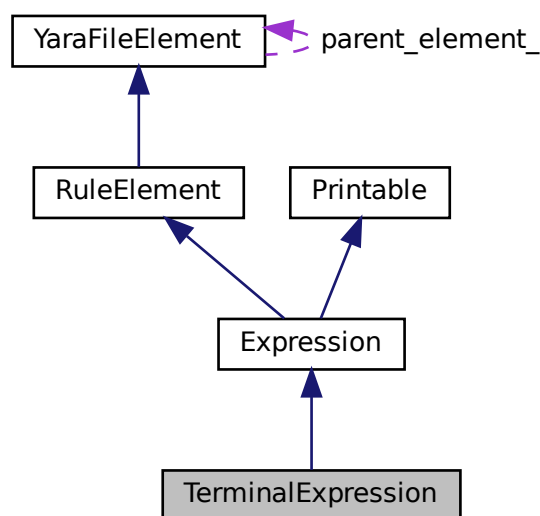
Base class for expression, that have not any subexpressions (operands)

```
#include <expression.h>
```

Inheritance diagram for TerminalExpression:



Collaboration diagram for TerminalExpression:



Public Member Functions

- bool [isValid](#) (std::string_view &msg) const override
- bool [areOperandsValid](#) () const override

Additional Inherited Members

5.152.1 Detailed Description

Base class for expression, that have not any subexpressions (operands)

5.152.2 Member Function Documentation

5.152.2.1 [areOperandsValid\(\)](#)

```
bool TerminalExpression::areOperandsValid ( ) const [override], [virtual]
```

Returns always true (this type of expression does not have any operands)

Implements [Expression](#).

5.152.2.2 [isValid\(\)](#)

```
bool TerminalExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Returns always true (terminal node can itself cannot be semantically invalid)

Implements [Expression](#).

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

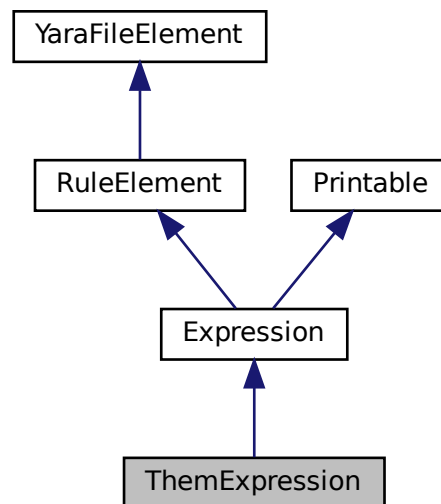
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.153 ThemExpression Class Reference

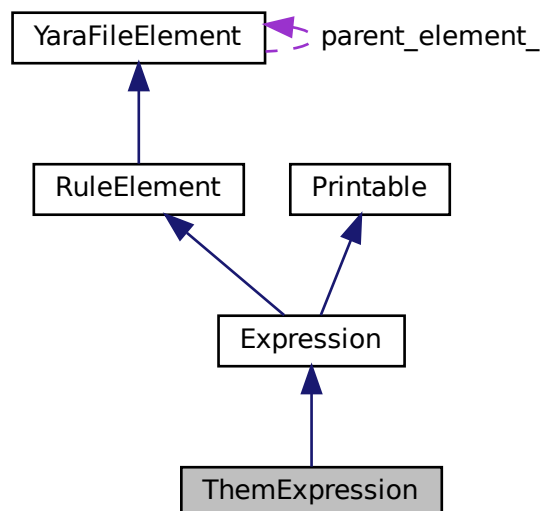
Represents 'them' keyword.

```
#include <expression.h>
```

Inheritance diagram for ThemExpression:



Collaboration diagram for ThemExpression:



Public Member Functions

- bool [isValid](#) (std::string_view &) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- [Expression::Type](#) [getType](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- void [accept](#) ([Visitor](#) *v) override

Additional Inherited Members

5.153.1 Detailed Description

Represents 'them' keyword.

5.153.2 Member Function Documentation

5.153.2.1 [accept\(\)](#)

```
void ThemExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.153.2.2 [areOperandsValid\(\)](#)

```
bool ThemExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.153.2.3 [getTextFormatted\(\)](#)

```
std::stringstream ThemExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.153.2.4 getType()

```
Expression::Type ThemExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.153.2.5 isComplete()

```
bool ThemExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.153.2.6 isValid()

```
bool ThemExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

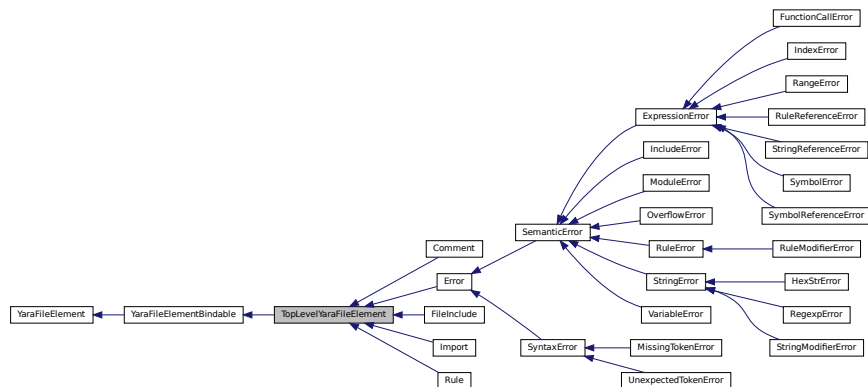
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.154 TopLevelYaraFileElement Class Reference

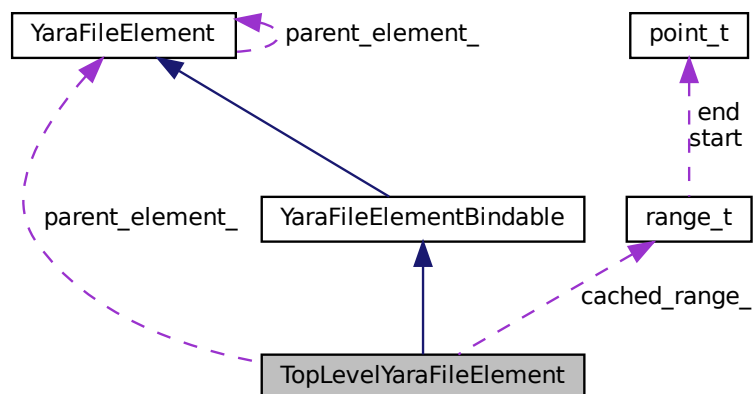
Class for top level yara file elements.

```
#include <yara_file_element.h>
```

Inheritance diagram for TopLevelYaraFileElement:



Collaboration diagram for TopLevelYaraFileElement:



Public Member Functions

- **TopLevelYaraFileElement** (const offset_t &offset, const size_t &len)
- [point_t](#) [getPosition](#) ()
- [range_t](#) [getRange](#) ()
- void [invalidateCache](#) ()
- void [setRangeCache](#) (const [range_t](#) &range)
- void [setPositionCache](#) (const [point_t](#) &position)
- bool [hasValidCache](#) ()

Protected Attributes

- `YaraFileElement * parent_element_ = nullptr`
Parent element that fixes elements on its place.
- `range_t cached_range_ = {{0, 0}, {0, 0}}`
Range cache to avoid unnecessary calls of offsetToPoint.
- `bool is_range_cache_valid_ = false`
- `bool is_cached_position_valid_ = false`

5.154.1 Detailed Description

Class for top level yara file elements.

Top level from point of view of semantic interface, so top level elements are rules, imports, includes but also comments and errors

5.154.2 Member Function Documentation

5.154.2.1 getPosition()

```
point_t TopLevelYaraFileElement::getPosition ( )
```

Returns position of first character of the yara file element

5.154.2.2 getRange()

```
range_t TopLevelYaraFileElement::getRange ( )
```

Returns [Range](#) where is [YaraFileElement](#) located

5.154.2.3 hasValidCache()

```
bool TopLevelYaraFileElement::hasValidCache ( )
```

Checks if cached values of position and range are valid

5.154.2.4 invalidateCache()

```
void TopLevelYaraFileElement::invalidateCache ( )
```

Invalidates cached range and position

5.154.2.5 setRangeCache()

```
void TopLevelYaraFileElement::setRangeCache (
    const range\_t & range )
```

Sets the range of rule, it is stored in range cache of the rule

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

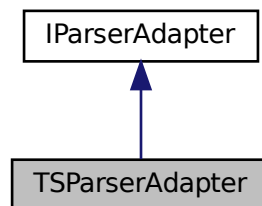
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file_element.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file_element.cpp](#)

5.155 TSParserAdapter Class Reference

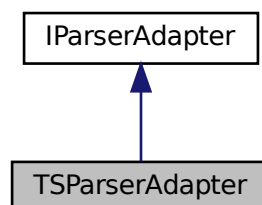
Class responsible for parsing of input string and converting TS structure to high level representation.

```
#include <parser_adapter.h>
```

Inheritance diagram for TSParserAdapter:



Collaboration diagram for TSParserAdapter:



Public Member Functions

- **TSParserAdapter** (std::shared_ptr< [YaramodConfig](#) > config)
- YaraSourcePtr [createYaraSource](#) (const std::string &string) override
- YaraSourcePtr **createYaraSource** (const std::string &string, const std::string &entry_file_path) override
- void [updateYaraSource](#) (const std::string &string, [YaraSource](#) *old_src) override
- void **updateYaraSource** (const std::string &string, const std::string &entry_file_path, [YaraSource](#) *old_src) override
- YaraSourcePtr [createYaraSourceFromFile](#) (std::string_view path) override
- void [updateYaraSourceFromFile](#) (std::string_view path, [YaraSource](#) *old_src) override
- YaraFilePtr [parse](#) ([YaraSource](#) *yara_src, const std::string &string, const std::string &name={})
- void [reparse](#) ([YaraSource](#) *parent_src, const std::string &string, const YaraFilePtr &old_yara_file)
- YaraFilePtr **parseFile** ([YaraSource](#) *parent_src, std::string_view path)
- void **reparseFile** ([YaraSource](#) *parent_src, std::string_view path, const YaraFilePtr &old_yara_file)

Additional Inherited Members

5.155.1 Detailed Description

Class responsible for parsing of input string and converting TS structure to high level representation.

5.155.2 Member Function Documentation

5.155.2.1 [createYaraSource\(\)](#)

```
YaraSourcePtr TSParserAdapter::createYaraSource (
    const std::string & string ) [override], [virtual]
```

Creates [YaraSource](#) object from the given string

Parameters

<i>string</i>	String in YARA language that should be parsed
---------------	---

Returns

Unique ptr to [YaraSource](#)

Implements [IParserAdapter](#).

5.155.2.2 [createYaraSourceFromFile\(\)](#)

```
YaraSourcePtr TSParserAdapter::createYaraSourceFromFile (
    std::string_view path ) [override], [virtual]
```

Creates [YaraSource](#) object from the file

Parameters

<i>path</i>	Path leading to file in YARA language that should be parsed
-------------	---

Returns

Unique ptr to [YaraSource](#)

Implements [IParserAdapter](#).

5.155.2.3 parse()

```
YaraFilePtr TSParserAdapter::parse (
    YaraSource * yara_src,
    const std::string & string,
    const std::string & name = {} )
```

Performs parsing of string and creates its high level representation

Performs parsing of string creates its high level representation

5.155.2.4 reparse()

```
void TSParserAdapter::reparse (
    YaraSource * parent_src,
    const std::string & string,
    const YaraFilePtr & old_yara_file )
```

Performs reparsing of string and and updates its high level representation

Performs reparsing of string and updates its high level representation < Set activity to true, because reparsing of file means, that it is still referenced from somewhere

< Update string

< Update tree and high level representation until there are any changes (to handle semantic errors)

5.155.2.5 updateYaraSource()

```
void TSParserAdapter::updateYaraSource (
    const std::string & string,
    YaraSource * old_src ) [override], [virtual]
```

Updates [YaraSource](#) object, that were created from the string

Parameters

<i>string</i>	String in YARA language that should be reparsed
<i>old_src</i>	Old YaraSource object, that should be updated

Implements [IParserAdapter](#).

5.155.2.6 updateYaraSourceFromFile()

```
void TSParserAdapter::updateYaraSourceFromFile (
    std::string_view path,
    YaraSource * old_src ) [override], [virtual]
```

Updates [YaraSource](#) object, that were created from the file

Note

Path can differ from the original path

Implements [IParserAdapter](#).

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

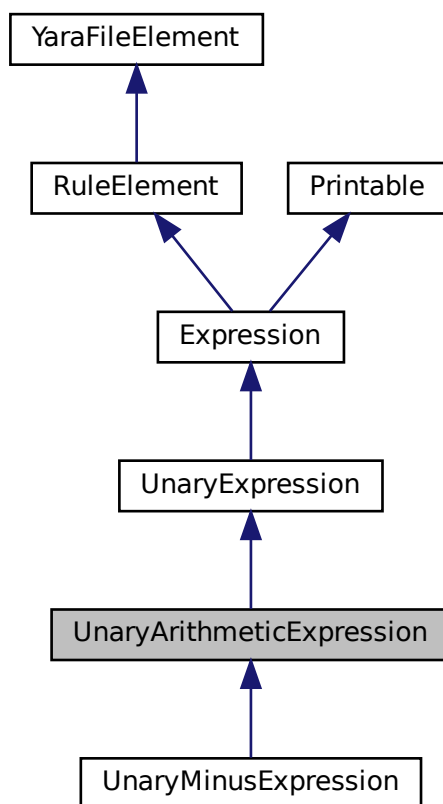
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[parser_adapter.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/[parser_adapter.cpp](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/[parser_adapter_expression.cpp](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/[parser_adapter_rule.cpp](#)

5.156 UnaryArithmeticExpression Class Reference

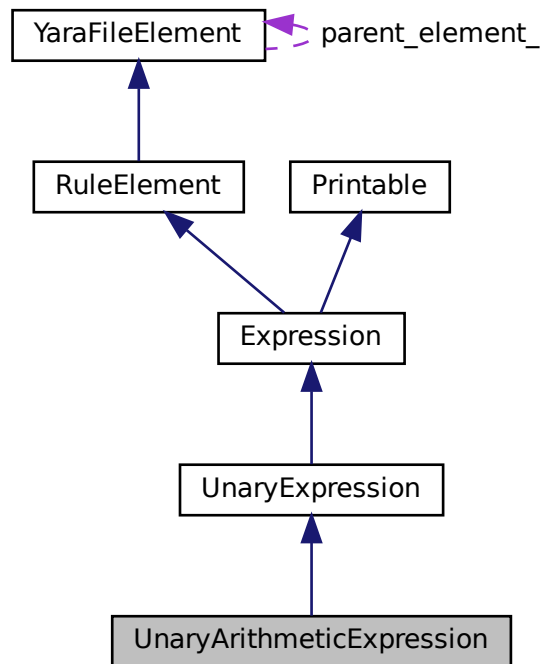
Base class for all arithmetic operators (expressions), which have one operand.

```
#include <expression.h>
```

Inheritance diagram for UnaryArithmeticExpression:



Collaboration diagram for UnaryArithmeticExpression:



Public Member Functions

- **UnaryArithmeticExpression** (ExpressionPtr op)
- bool `isValid` (std::string_view &msg) const override
- `Expression::Type` `getType` () const override

Additional Inherited Members

5.156.1 Detailed Description

Base class for all arithmetic operators (expressions), which have one operand.

5.156.2 Member Function Documentation

5.156.2.1 `getType()`

```
Expression::Type UnaryArithmeticExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements `Expression`.

5.156.2.2 isValid()

```
bool UnaryArithmeticExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

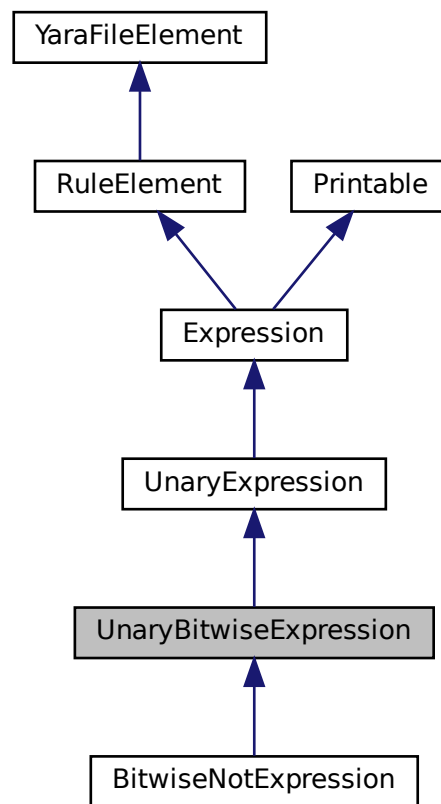
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.157 UnaryBitwiseExpression Class Reference

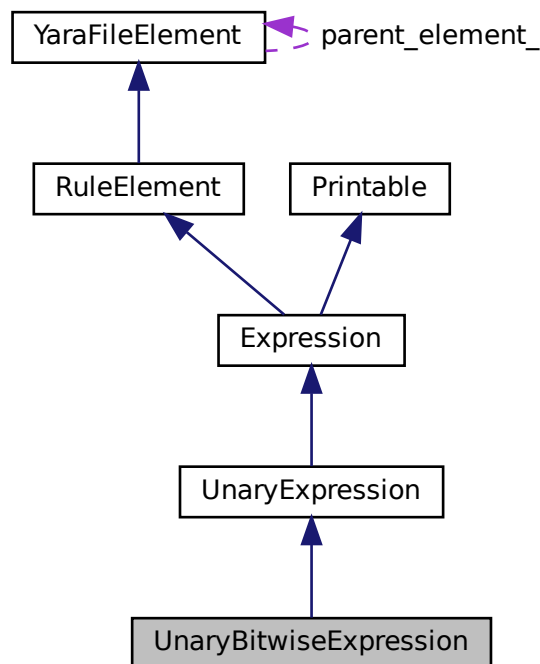
Base class for bitwise unary operators (expressions), that have one operand. For example bitwise not (~) operator.

```
#include <expression.h>
```

Inheritance diagram for UnaryBitwiseExpression:



Collaboration diagram for UnaryBitwiseExpression:



Public Member Functions

- **UnaryBitwiseExpression** (ExpressionPtr op)
- bool [isValid](#) (std::string_view &msg) const override
- [Expression::Type](#) [getType](#) () const override

Additional Inherited Members

5.157.1 Detailed Description

Base class for bitwise unary operators (expressions), that have one operand. For example bitwise not (`~`) operator.

5.157.2 Member Function Documentation

5.157.2.1 getType()

```
Expression::Type UnaryBitwiseExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.157.2.2 isValid()

```
bool UnaryBitwiseExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

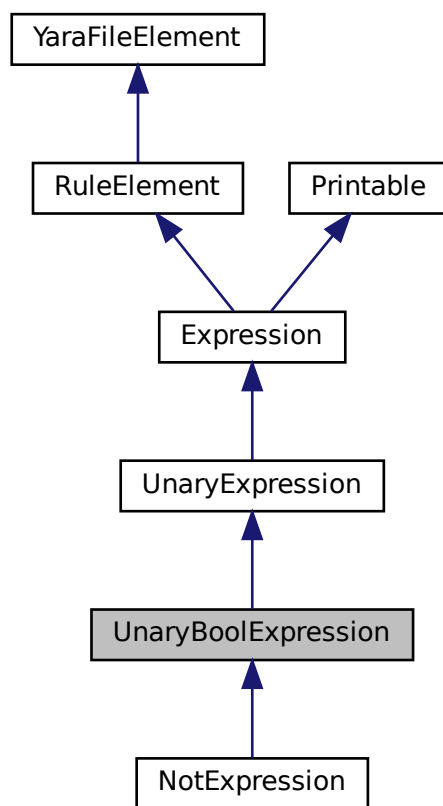
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.158 UnaryBoolExpression Class Reference

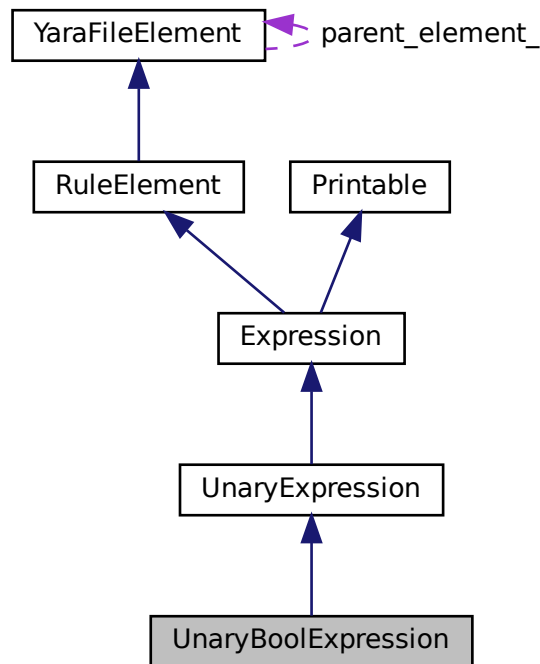
Base class for binary operators, that need boolean operands. For example and, or.

```
#include <expression.h>
```

Inheritance diagram for UnaryBoolExpression:



Collaboration diagram for UnaryBoolExpression:



Public Member Functions

- **UnaryBoolExpression** (ExpressionPtr op)
- bool `isValid` (std::string_view &msg) const override
- `Expression::Type` `getType` () const override

Additional Inherited Members

5.158.1 Detailed Description

Base class for binary operators, that need boolean operands. For example and, or.

5.158.2 Member Function Documentation

5.158.2.1 `getType()`

```
Expression::Type UnaryBoolExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.158.2.2 isValid()

```
bool UnaryBoolExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

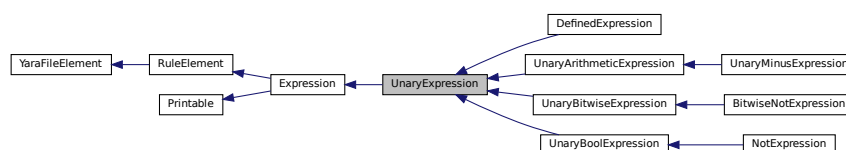
- /home/vojtech.dvorak1/Documents/yaromod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaromod-v4/src/rule/[expression.cpp](#)

5.159 UnaryExpression Class Reference

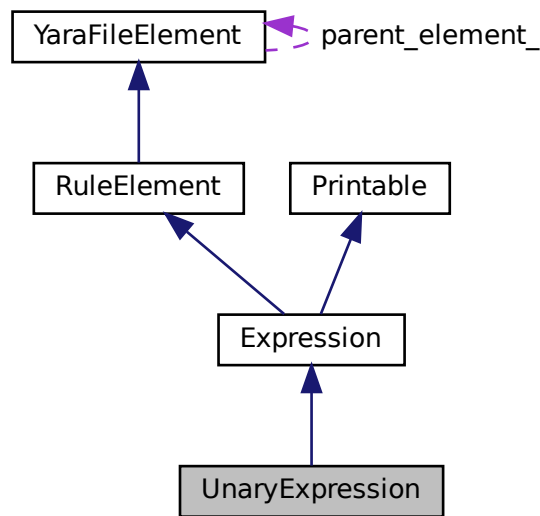
Base class for unary operators (such as unary minus, negation...).

```
#include <expression.h>
```

Inheritance diagram for UnaryExpression:



Collaboration diagram for UnaryExpression:



Public Member Functions

- **UnaryExpression** (ExpressionPtr op)
- bool [isComplete](#) () const override
- bool [areOperandsValid](#) () const override
- virtual std::string **opsign** () const =0
- std::stringstream [getTextFormatted](#) () const override
- const ExpressionPtr & **getOp** () const

Protected Attributes

- ExpressionPtr **op_** = nullptr

Additional Inherited Members

5.159.1 Detailed Description

Base class for unary operators (such as unary minus, negation...).

Similar to Binary expression ([BinaryExpression](#))

5.159.2 Member Function Documentation

5.159.2.1 areOperandsValid()

```
bool UnaryExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.159.2.2 getTextFormatted()

```
std::stringstream UnaryExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.159.2.3 isComplete()

```
bool UnaryExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

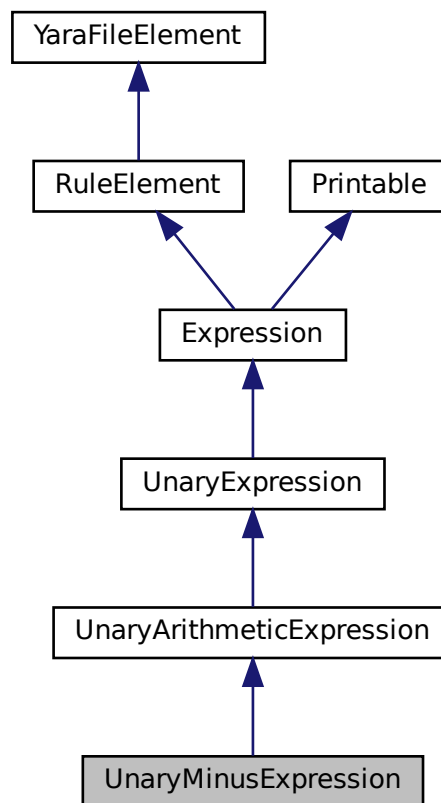
Implements [Expression](#).

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

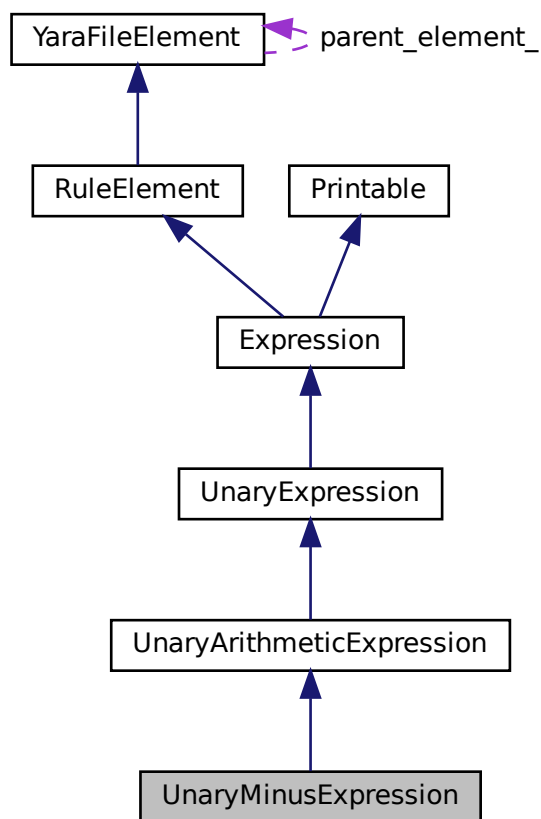
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp](#)

5.160 UnaryMinusExpression Class Reference

Inheritance diagram for UnaryMinusExpression:



Collaboration diagram for UnaryMinusExpression:



Public Member Functions

- **UnaryMinusExpression** (ExpressionPtr op)
- `std::string opsign ()` const override
- `void accept (Visitor *v)` override

Additional Inherited Members

5.160.1 Member Function Documentation

5.160.1.1 accept()

```
void UnaryMinusExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

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

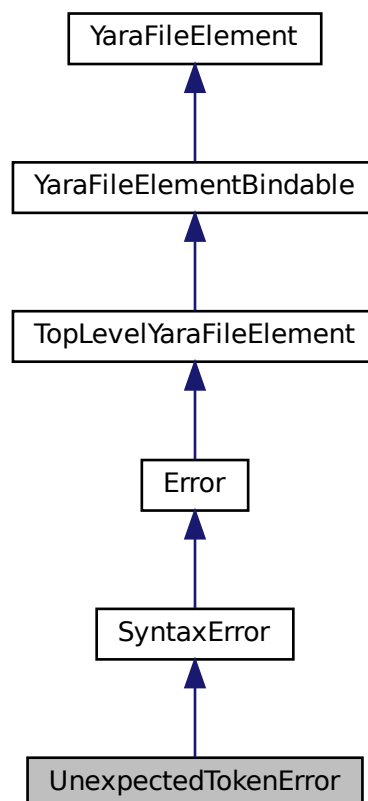
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h](#)

5.161 UnexpectedTokenError Class Reference

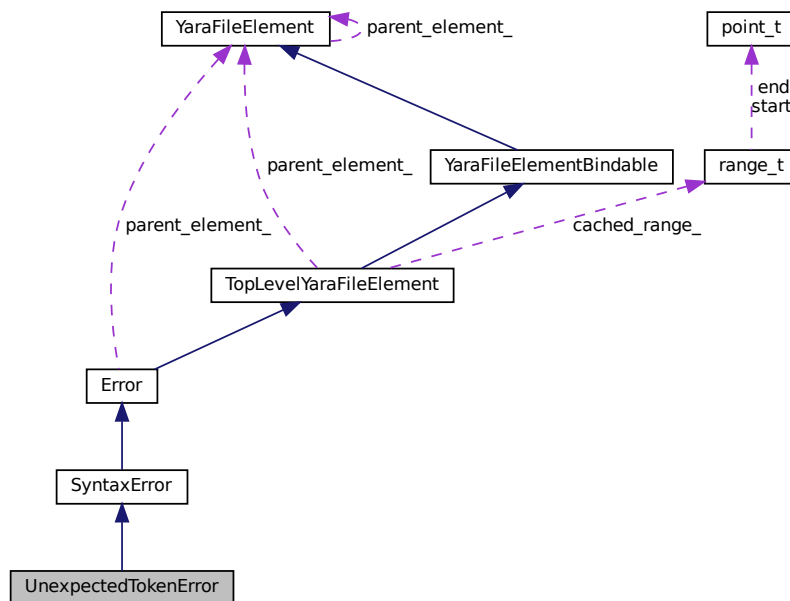
Class for syntax error caused by unexpected token.

```
#include <error.h>
```

Inheritance diagram for UnexpectedTokenError:



Collaboration diagram for UnexpectedTokenError:



Public Member Functions

- **UnexpectedTokenError** (const std::string &token, const offset_t &offset, const size_t &len)
- [MissingTokenErrorException](#) exception ()

Additional Inherited Members

5.161.1 Detailed Description

Class for syntax error caused by unexpected token.

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

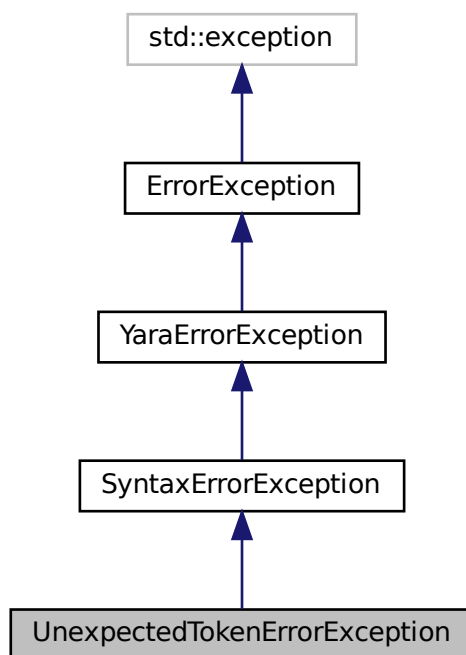
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.162 UnexpectedTokenErrorException Class Reference

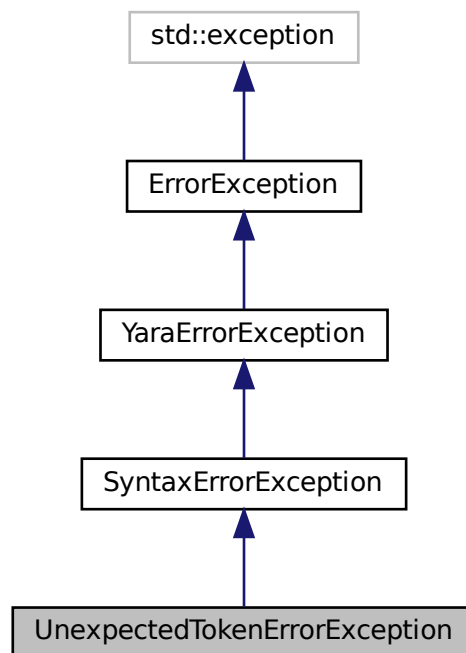
Exception representing syntax error in parsed source code.

```
#include <error.h>
```

Inheritance diagram for UnexpectedTokenErrorException:



Collaboration diagram for UnexpectedTokenErrorException:



Public Member Functions

- **UnexpectedTokenErrorException** (offset_t offset, size_t len, std::string msg={})

5.162.1 Detailed Description

Exception representing syntax error in parsed source code.

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.163 var_def_t Struct Reference

Structure, that hold all information about in-expression defined variable.

```
#include <expression.h>
```


Public Attributes

- `std::string id`
Identifier of variable.
- `Symbol::Type type = Symbol::Type::Unknown`
Type of variable.
- `Expression::Type dtype = Expression::Type::Undefined`
Data type of variables.
- `const json * module_ctx = nullptr`
JSON context of variable.

5.163.1 Detailed Description

Structure, that hold all information about in-expression defined variable.

In case of YARA, it is possible to define variables only in 'for' expression

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

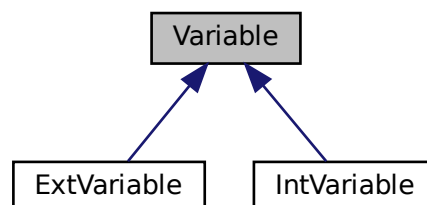
- </home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h>

5.164 Variable Class Reference

Base class for variables.

```
#include <variable.h>
```

Inheritance diagram for Variable:



Public Member Functions

- **Variable** (const std::string &id)
- const std::string & [getId](#) () const
- void [setId](#) (std::string &&id)
- void [setId](#) (const std::string &id)

Protected Attributes

- `std::string id_ = std::string({})`

5.164.1 Detailed Description

Base class for variables.

5.164.2 Member Function Documentation

5.164.2.1 getId()

```
const std::string & Variable::getId ( ) const
```

Returns id of variable

5.164.2.2 setId()

```
void Variable::setId (
    std::string && id )
```

Sets the id of variable

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

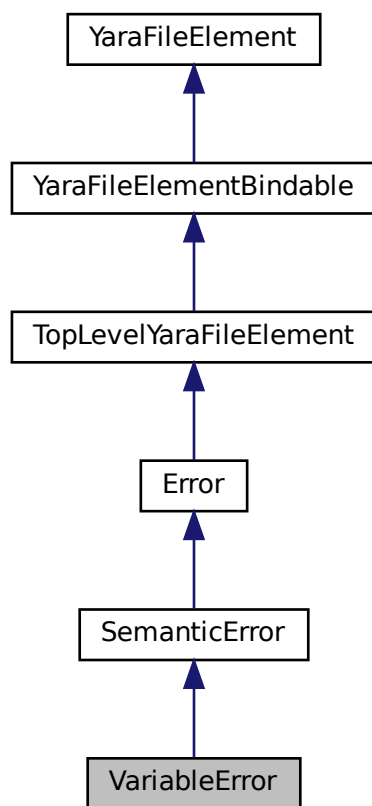
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/variable.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/variable.cpp](#)

5.165 VariableError Class Reference

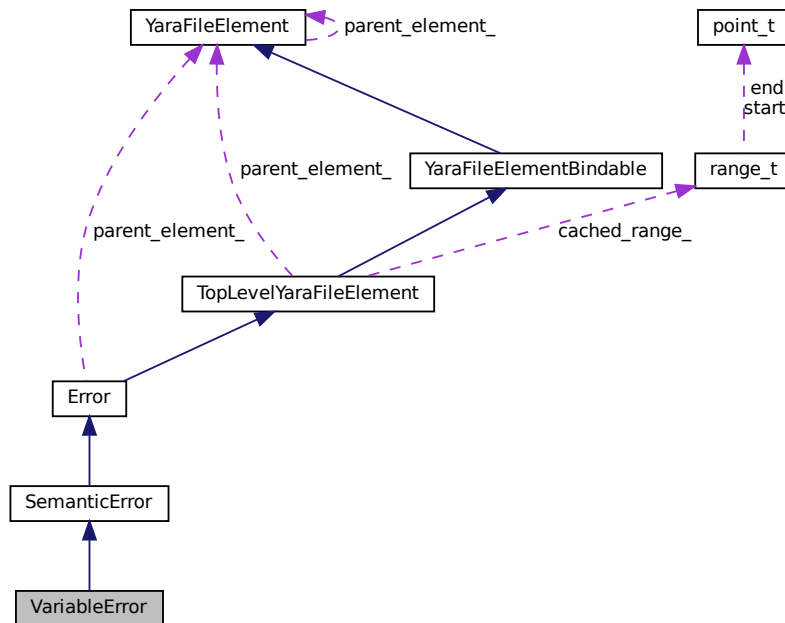
[Error](#) of variable definition.

```
#include <error.h>
```

Inheritance diagram for VariableError:



Collaboration diagram for VariableError:



Public Member Functions

- **VariableError** (const std::string &desc, const offset_t &offset, const size_t &len)
- [VariableErrorException](#) exception ()

Additional Inherited Members

5.165.1 Detailed Description

[Error](#) of variable definition.

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

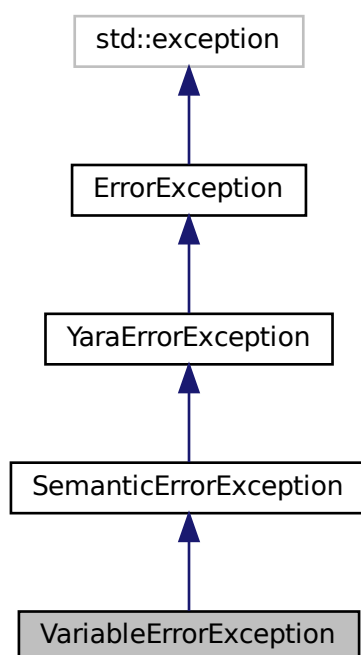
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.166 VariableErrorException Class Reference

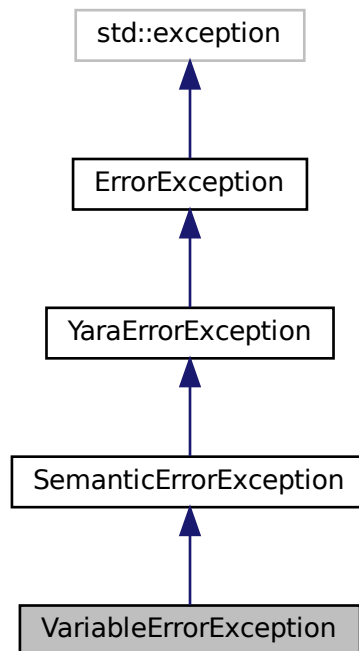
Exception for errors in variable definition.

```
#include <error.h>
```

Inheritance diagram for VariableErrorException:



Collaboration diagram for VariableErrorException:



Public Member Functions

- **VariableErrorException** (offset_t offset, size_t len, std::string msg={})

5.166.1 Detailed Description

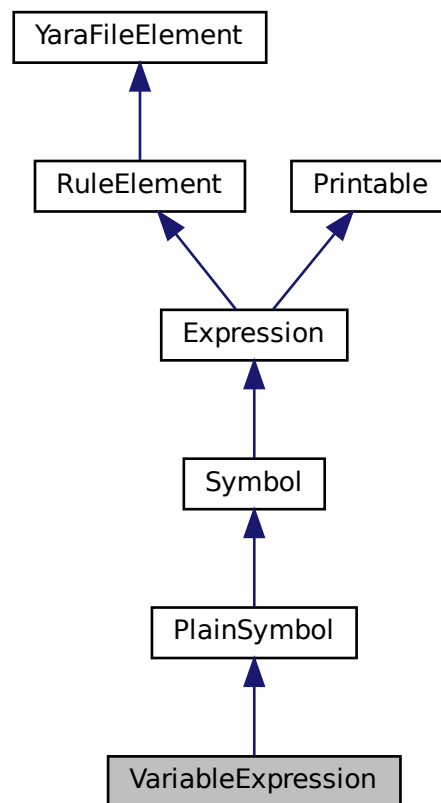
Exception for errors in variable definition.

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

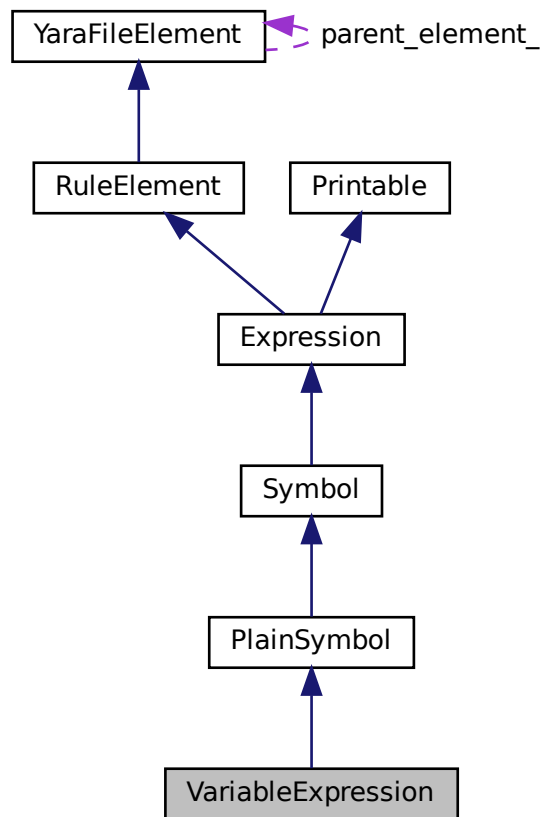
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.167 VariableExpression Class Reference

Inheritance diagram for VariableExpression:



Collaboration diagram for VariableExpression:



Public Member Functions

- **VariableExpression** (std::string_view id)
- void **accept** (Visitor *v) override

Additional Inherited Members

5.167.1 Member Function Documentation

5.167.1.1 accept()

```
void VariableExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Reimplemented from [PlainSymbol](#).

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

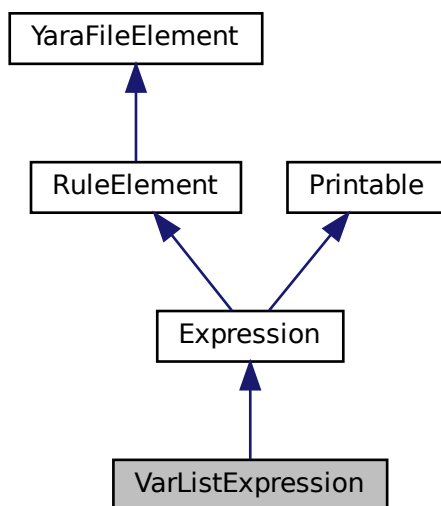
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)

5.168 VarListExpression Class Reference

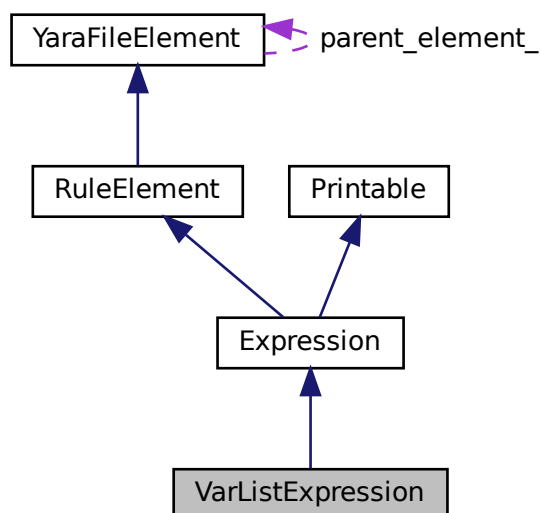
Represents list of variables, that are defined in 'for' expression for usage inside internal expression of 'for'.

```
#include <expression.h>
```

Inheritance diagram for VarListExpression:



Collaboration diagram for VarListExpression:



Public Member Functions

- **VarListExpression** (std::vector< std::string > &&vars)
- bool [isValid](#) (std::string_view &) const override
- bool [areOperandsValid](#) () const override
- bool [isComplete](#) () const override
- std::stringstream [getTextFormatted](#) () const override
- [Expression::Type](#) [getType](#) () const override
- void [accept](#) ([Visitor](#) *v) override
- const std::vector< std::string > & [getVariables](#) () const

Additional Inherited Members

5.168.1 Detailed Description

Represents list of variables, that are defined in 'for' expression for usage inside internal expression of 'for'.

5.168.2 Member Function Documentation

5.168.2.1 [accept\(\)](#)

```
void VarListExpression::accept (
    Visitor * v ) [inline], [override], [virtual]
```

Accept method for visitor pattern

Implements [Expression](#).

5.168.2.2 [areOperandsValid\(\)](#)

```
bool VarListExpression::areOperandsValid ( ) const [override], [virtual]
```

Checks if operands are valid. This method is used to determine, whether if semantic error appears in current expression or in its subexpressions (operands)

Implements [Expression](#).

5.168.2.3 getTextFormatted()

```
std::stringstream VarListExpression::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.168.2.4 getType()

```
Expression::Type VarListExpression::getType ( ) const [override], [virtual]
```

Returns the data type of expression

Implements [Expression](#).

5.168.2.5 isComplete()

```
bool VarListExpression::isComplete ( ) const [override], [virtual]
```

Checks whther expression is complete - if it has all mandatory operands.

Implements [Expression](#).

5.168.2.6 isValid()

```
bool VarListExpression::isValid (
    std::string_view & msg ) const [override], [virtual]
```

Checks the validity of expression - data types of operands are typically checked and their combinations (checks the semantic of expression). It is called recursively for subexpressions of expression (e. g. operands) so semantic errors are propagated to the top level expression.

Parameters

<i>msg</i>	error message, that describes an error
------------	--

Returns

true if expression (and its subexpressions) is valid

Implements [Expression](#).

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

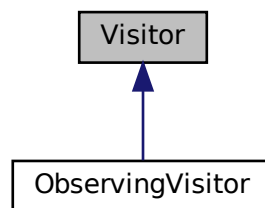
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[expression.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/[expression.cpp](#)

5.169 Visitor Class Reference

Virtual base class for concrete visitor classes.

```
#include <visitor.h>
```

Inheritance diagram for Visitor:



Public Member Functions

- virtual void **visitRange** ([Range](#) *r)=0
- virtual void **visitEnum** ([Enum](#) *e)=0
- virtual void **visitParentheses** ([ParenthesesExpression](#) *e)=0
- virtual void **visitIn** ([InExpression](#) *e)=0
- virtual void **visitMatches** ([MatchesExpression](#) *e)=0
- virtual void **visitAt** ([AtExpression](#) *e)=0
- virtual void **visitAdd** ([AddExpression](#) *e)=0
- virtual void **visitSub** ([SubExpression](#) *e)=0
- virtual void **visitMul** ([MulExpression](#) *e)=0
- virtual void **visitRemainder** ([RemainderExpression](#) *e)=0
- virtual void **visitDiv** ([DivExpression](#) *e)=0
- virtual void **visitLeftShift** ([LeftShiftExpression](#) *e)=0
- virtual void **visitRightShift** ([RightShiftExpression](#) *e)=0
- virtual void **visitBitwiseOr** ([BitwiseOrExpression](#) *e)=0
- virtual void **visitBitwiseAnd** ([BitwiseAndExpression](#) *e)=0
- virtual void **visitBitwiseXor** ([BitwiseXorExpression](#) *e)=0
- virtual void **visitLt** ([LtExpression](#) *e)=0

- virtual void **visitLte** ([LteExpression](#) *e)=0
- virtual void **visitGt** ([GtExpression](#) *e)=0
- virtual void **visitGte** ([GteExpression](#) *e)=0
- virtual void **visitEq** ([EqExpression](#) *e)=0
- virtual void **visitNeq** ([NeqExpression](#) *e)=0
- virtual void **visitContains** ([ContainsExpression](#) *e)=0
- virtual void **visitIContains** ([IContainsExpression](#) *e)=0
- virtual void **visitStartsWith** ([StartsWithExpression](#) *e)=0
- virtual void **visitIStartsWith** ([IStartsWithExpression](#) *e)=0
- virtual void **visitEndsWith** ([EndsWithExpression](#) *e)=0
- virtual void **visitIEndsWith** ([IEndsWithExpression](#) *e)=0
- virtual void **visitIEq** ([IEqExpression](#) *e)=0
- virtual void **visitAnd** ([AndExpression](#) *e)=0
- virtual void **visitOr** ([OrExpression](#) *e)=0
- virtual void **visitDefined** ([DefinedExpression](#) *e)=0
- virtual void **visitUnaryMinus** ([UnaryMinusExpression](#) *e)=0
- virtual void **visitBitwiseNot** ([BitwiseNotExpression](#) *e)=0
- virtual void **visitNot** ([NotExpression](#) *e)=0
- virtual void **visitLiteral** ([LiteralExpression](#) *e)=0
- virtual void **visitSize** ([SizeExpression](#) *e)=0
- virtual void **visitRegexp** ([RegexpExpression](#) *e)=0
- virtual void **visitString** ([StringExpression](#) *e)=0
- virtual void **visitStringWildcard** ([StringWildcardExpression](#) *e)=0
- virtual void **visitStringCount** ([StringCountExpression](#) *e)=0
- virtual void **visitStringOffset** ([StringOffsetExpression](#) *e)=0
- virtual void **visitStringMatchLength** ([StringMatchLengthExpression](#) *e)=0
- virtual void **visitPlainSymbol** ([PlainSymbol](#) *symbol)=0
- virtual void **visitArray** ([ArrayExpression](#) *e)=0
- virtual void **visitStruct** ([StructExpression](#) *e)=0
- virtual void **visitFunctionCall** ([FunctionCallExpression](#) *e)=0
- virtual void **visitVariable** ([VariableExpression](#) *e)=0
- virtual void **visitRuleWildcard** ([RuleWildcardExpression](#) *e)=0
- virtual void **visitNone** ([NoneExpression](#) *e)=0
- virtual void **visitAny** ([AnyExpression](#) *e)=0
- virtual void **visitAll** ([AllExpression](#) *e)=0
- virtual void **visitPercentQuantifier** ([PercentQuantifierExpression](#) *e)=0
- virtual void **visitThem** ([ThemExpression](#) *e)=0
- virtual void **visitStringSet** ([StringSetExpression](#) *e)=0
- virtual void **visitRuleSet** ([RuleSetExpression](#) *e)=0
- virtual void **visitExpressionSet** ([ExpressionSetExpression](#) *e)=0
- virtual void **visitOf** ([OfExpression](#) *e)=0
- virtual void **visitFor** ([ForExpression](#) *e)=0
- virtual void **visitForInt** ([ForIntExpression](#) *e)=0
- virtual void **visitVarList** ([VarListExpression](#) *e)=0

5.169.1 Detailed Description

Virtual base class for concrete visitor classes.

If it is used as base class for concrete visitor, it ensures, that all possible types of expressions will be handled during traversal of expressions.

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

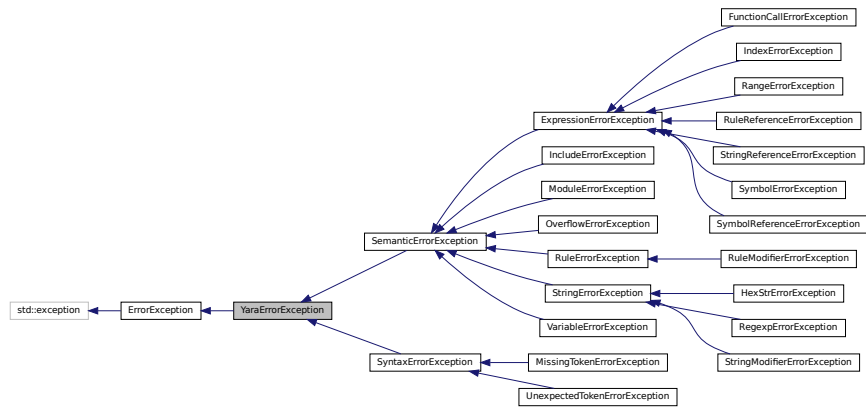
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/visitor.h](#)

5.170 YaraErrorException Class Reference

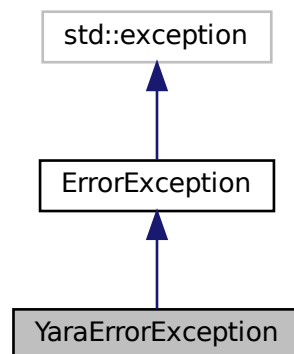
Base Exception class for errors located in input YARA code.

```
#include <error.h>
```

Inheritance diagram for YaraErrorException:



Collaboration diagram for YaraErrorException:



Public Member Functions

- **YaraErrorException** (offset_t offset, size_t len, std::string msg={})
- size_t **getLen** () const noexcept
- offset_t **getOffset** () const noexcept
- [range_t](#) **getRange** (std::string_view str) const noexcept

5.170.1 Detailed Description

Base Exception class for errors located in input YARA code.

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

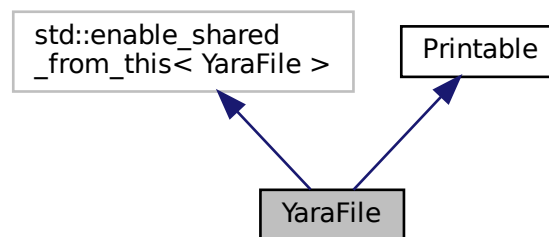
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/error.cpp](#)

5.171 YaraFile Class Reference

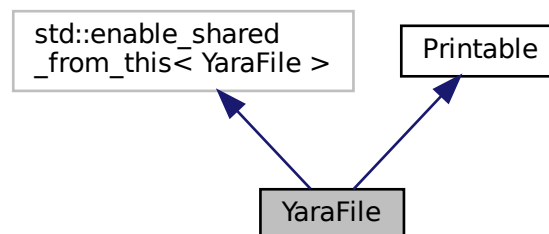
Represents on continuous sequence of yara rules, file includes and imports.

```
#include <yara_file.h>
```

Inheritance diagram for YaraFile:



Collaboration diagram for YaraFile:



Public Member Functions

- **YaraFile** ([YaraSource](#) *parent_src)
- const std::string & [getName](#) () const
- void [setName](#) (const std::string &new_name)
- void **setName** (std::string_view new_name)
- template<class T, class E >
E * [addError](#) ([ErrorCollector](#)< T > &error_collector, std::unique_ptr< E > &&err)
- const [SymTabOffset](#)< [SyntaxError](#) > & [getSyntaxErrors](#) () const
- void [clearSyntaxErrors](#) ()
- template<class E >
E * [addSyntaxError](#) (std::unique_ptr< E > &&err)
- template<class E >
E * **addSyntaxError** (const std::string &desc, const offset_t &offset, size_t len)
- template<class E >
E * **addSyntaxError** (const std::string &desc, const offset_t &offset, size_t len, const [range_t](#) &range)
- template<typename Fn >
uint32_t [removeSyntaxErrors](#) (const Fn &f)
- template<typename Fn >
void [updateSyntaxErrors](#) (const Fn &f)
- void [dumpSyntaxErrors](#) ()
- const [SymTabOffset](#)< [SemanticError](#) > & [getSemanticErrors](#) () const
- void [clearSemanticErrors](#) ()
- template<class E >
E * [addSemanticError](#) (std::unique_ptr< E > &&err)
- template<class E >
E * **addSemanticError** (std::string_view desc, const offset_t &offset, size_t len)
- template<class E >
E * **addSemanticError** (std::string_view desc, const offset_t &offset, size_t len, const [range_t](#) &range)
- template<typename Fn >
uint32_t [removeSemanticErrors](#) (const Fn &f)
- template<typename Fn >
void **updateSemanticErrors** (const Fn &f)
- void **dumpSemanticErrors** () const
- [Rule](#) * [ruleLookUp](#) (const std::string &rule_id, std::vector< [YaraFileElementBindable](#) * > &includes)
- [RuleError](#) * [createRuleError](#) (const std::string &err_msg, [Rule](#) *wrong_rule, const std::vector< [YaraFileElementBindable](#) * > &includes)
- [Rule](#) * [addRule](#) ([RulePtr](#) &&rule)
- void [clearLocalRules](#) ()
- const [SymTab](#)< [Rule](#) > & [getLocalRules](#) () const
- std::vector< [Rule](#) * > [getRules](#) () const
- std::vector< [Rule](#) * > [getRules](#) (const std::string &name) const
- bool [hasRule](#) (std::string_view name) const
- bool [hasRule](#) (std::string_view name, offset_t before) const
- [Rule](#) * [getRule](#) (std::string_view name) const
- [Rule](#) * **getRule** (std::string_view name, offset_t before) const
- [Rule](#) * [getRule](#) (std::string_view name, offset_t before, std::vector< [YaraFileElementBindable](#) * > &includes, bool make_lookup_from_top=false) const
- [Rule](#) * [getRule](#) (std::string_view name, const [YaraFile](#) *stop_file, std::vector< [YaraFileElementBindable](#) * > &includes)
- const [RulePtr](#) & [getLocalRule](#) (const std::string &name) const
- [Rule](#) * **getLocalRule** (uint32_t index) const
- [Rule](#) * [getLocalRuleByRow](#) (uint32_t row)
- template<typename Fn >
uint32_t [removeLocalRules](#) (const Fn &f)

- template<typename Fn >
void [updateLocalRules](#) (const Fn &f)
- void [checkDuplicatedRules](#) (const YaraFilePtr &other, [FileInclude](#) *include)
- [FileInclude](#) * [addFileInclude](#) (std::unique_ptr< [FileInclude](#) > include)
- void [clearLocalFileIncludes](#) ()
- [FileInclude](#) * [getLocalFileInclude](#) (uint32_t index) const
- const [SymTab](#)< [FileInclude](#) > & [getLocalFileIncludes](#) () const
- std::vector< [FileInclude](#) * > [getFileIncludes](#) () const
- template<typename Fn >
uint32_t [removeLocalFileIncludes](#) (const Fn &f)
- template<typename Fn >
void [updateLocalFileIncludes](#) (const Fn &f)
- [Import](#) * [getImport](#) (std::string_view name) const
- [Import](#) * [getImport](#) (std::string_view name, offset_t before) const
- [Import](#) * [getImport](#) (std::string_view name, offset_t before, std::vector< [YaraFileElementBindable](#) * > &includes, bool make_lookup_from_top=false) const
- [Import](#) * [getImport](#) (std::string_view name, const [YaraFile](#) *stop_file, std::vector< [YaraFileElementBindable](#) * > &includes)
- [Import](#) * [addImport](#) (std::unique_ptr< [Import](#) > import)
- void [clearLocalImports](#) ()
- [Import](#) * [getLocalImport](#) (uint32_t index) const
- std::vector< [Import](#) * > [getImports](#) () const
- const [SymTabDuplId](#)< [Import](#) > & [getLocalImports](#) () const
- template<typename Fn >
uint32_t [removeLocalImports](#) (const Fn &f)
- template<typename Fn >
void [updateLocalImports](#) (const Fn &f)
- void [addComment](#) (std::unique_ptr< [Comment](#) > comment)
- void [clearComments](#) ()
- const std::unique_ptr< [Comment](#) > & [getCommentByIndex](#) (uint32_t index) const
- const [SymTabOffset](#)< [Comment](#) > & [getComments](#) () const
- template<typename Fn >
uint32_t [removeComments](#) (const Fn &f)
- template<typename Fn >
void [updateComments](#) (const Fn &f)
- void [addWrong](#) (std::unique_ptr< [TopLevelYaraFileElement](#) > &&element)
- void [clearWrong](#) ()
- void [clearAll](#) ()
- void [addEdit](#) (const [edit_t](#) &edit)
- bool [hasFastEdit](#) ()
- void [edit](#) (const offset_t &offset, const size_t &ins_n, const size_t &del_n)
- void [edit](#) (const [range_t](#) &edited_range, const std::string &text)
- void [undo](#) ()
- void [clearEdits](#) ()
- std::vector< [edit_t](#) > & [getEdits](#) ()
- bool [wasEdited](#) ()
- bool [mustBeReparsed](#) ()
- void [removeGlobalErrors](#) ()
- [offset_range_t](#) [removeElementsInRange](#) (const offset_t &start, const offset_t &end)
- void [shiftElements](#) (const offset_t &start, const uint32_t &start_row, const int32_t &delta, const int32_t &row_delta, const int32_t &col_delta)
- std::vector< [offset_edit_range_t](#) > [modify](#) (TSTree *new_tree)
- void [setActivity](#) (bool state=true)
- bool [isActive](#) ()
- void [setErrorMode](#) ([YaramodConfig::ErrorMode](#) new_mode)

- void `notifyChange` (`YaraFileElement` *element)
- void `bind` (`YaraFileElement` *parent, `YaraFileElement` *dependency)
- void `bind` (const std::vector< `YaraFileElementBindable` * > &parents, `YaraFileElement` *dependency)
- void `deleteBindings` (`YaraFileElement` *element)
- void `moveBindings` (`YaraFileElement` *original_parent, `YaraFileElementBindable` *new_parent)
- void `checkRange` (const `offset_range_t` &range_to_be_checked)
- void `clearToBeChecked` ()
- `FileContext` & `ctx` ()
- bool `isIsolated` () const
- void `setIsolated` (bool new_state=true)
- void `addMissingSymbol` (const std::string &missing, const `offset_range_t` &range)
- void `removeMissingSymbol` (const std::string &missing)
- void `checkMissing` (const std::string &new_symbol)
- void `notifyAddedSymbol` (const std::string &id)
- std::stringstream `getTextFormatted` () const override

5.171.1 Detailed Description

Represents on continuous sequence of yara rules, file includes and imports.

Also works as stateful object, that holds all necessary data for reparsing.

5.171.2 Member Function Documentation

5.171.2.1 `addComment()`

```
void YaraFile::addComment (
    std::unique_ptr< Comment > comment )
```

Adds comment to the tab of the comments

5.171.2.2 `addEdit()`

```
void YaraFile::addEdit (
    const edit_t & edit )
```

Adds new edit to the edit buffer

5.171.2.3 `addError()`

```
template<class T , class E >
E * YaraFile::addError (
    ErrorCollector< T > & error_collector,
    std::unique_ptr< E > && err )
```

Adds error object to the specified error collector

Note

Helper template methods to avoid redundancies

5.171.2.4 addFileInclude()

```
FileInclude * YaraFile::addFileInclude (
    std::unique_ptr< FileInclude > include )
```

Adds [FileInclude](#) object to includes of yara file < There is already include with the same path

5.171.2.5 addImport()

```
Import * YaraFile::addImport (
    std::unique_ptr< Import > import )
```

Adds [Import](#) object to the tab with imports

5.171.2.6 addMissingSymbol()

```
void YaraFile::addMissingSymbol (
    const std::string & missing,
    const offset_range_t & range )
```

Add missing symbol to the map of the missing symbols

5.171.2.7 addRule()

```
Rule * YaraFile::addRule (
    RulePtr && rule )
```

Adds [Rule](#) object to the symbol tab < Redirect bindings of wrong rule to the correct rule

< Save the wrong rule to wrong element tab

5.171.2.8 addSemanticError()

```
template<class E >
E * YaraFile::addSemanticError (
    std::unique_ptr< E > && err )
```

Adds [SemanticError](#) to the yara file object

Returns

Raw pointer to the newly added semantic error

5.171.2.9 addSyntaxError()

```
template<class E >
E * YaraFile::addSyntaxError (
    std::unique_ptr< E > && err )
```

Adds [SyntaxError](#) to the yara file object

Returns

Raw pointer to the newly added [SyntaxError](#) object (to be additionally modified)

5.171.2.10 addWrong()

```
void YaraFile::addWrong (
    std::unique_ptr< TopLevelYaraFileElement > && element )
```

Adds element to tab with file elements

5.171.2.11 bind() [1/2]

```
void YaraFile::bind (
    const std::vector< YaraFileElementBindable * > & parents,
    YaraFileElement * dependency )
```

Creates binding between all elements in vector (they acts as parents) and dependency [YaraFileObject](#) < Create binding for all elements in vector

5.171.2.12 bind() [2/2]

```
void YaraFile::bind (
    YaraFileElement * parent,
    YaraFileElement * dependency )
```

Creates binding between parent element and dependent element

5.171.2.13 checkDuplicatedRules()

```
void YaraFile::checkDuplicatedRules (
    const YaraFilePtr & other,
    FileInclude * include )
```

Checks if this and other yara file have any rule with the same ids < Every rule should be at least once in [YaraFile](#) (or in included [YaraFile](#))

5.171.2.14 checkMissing()

```
void YaraFile::checkMissing (
    const std::string & new_symbol )
```

Converts ranges that belongs to specific missing symbols to ranges that will be checked during reparsing

5.171.2.15 checkRange()

```
void YaraFile::checkRange (
    const offset_range_t & range_to_be_checked )
```

Adds new range to to_be_checked_vector

5.171.2.16 clearAll()

```
void YaraFile::clearAll ( )
```

Clears all symbol tables and errors of yara file

5.171.2.17 clearComments()

```
void YaraFile::clearComments ( )
```

Clears the tab with comments

5.171.2.18 clearEdits()

```
void YaraFile::clearEdits ( )
```

Clears edit buffer

5.171.2.19 clearLocalFileIncludes()

```
void YaraFile::clearLocalFileIncludes ( )
```

Clears symtab with local file includes

5.171.2.20 clearLocalImports()

```
void YaraFile::clearLocalImports ( )
```

Clear tab with local imports

5.171.2.21 clearLocalRules()

```
void YaraFile::clearLocalRules ( )
```

Clears symbol tab with local rules < Remove rules from the list

5.171.2.22 clearSemanticErrors()

```
void YaraFile::clearSemanticErrors ( )
```

Clears tab with semantic errors in the current file

5.171.2.23 clearSyntaxErrors()

```
void YaraFile::clearSyntaxErrors ( )
```

Clears tab with syntax errors in the current file

5.171.2.24 clearToBeChecked()

```
void YaraFile::clearToBeChecked ( )
```

Clears to_be_checked_vector

5.171.2.25 clearWrong()

```
void YaraFile::clearWrong ( )
```

Clears tab with wrong file elements

5.171.2.26 createRuleError()

```
RuleError * YaraFile::createRuleError (
    const std::string & err_msg,
    Rule * wrong_rule,
    const std::vector< YaraFileElementBindable * > & includes )
```

Create correct duplicated rule message with the correct location < Rule is in another file

5.171.2.27 ctx()

```
FileContext & YaraFile::ctx ( )
```

Provides access to parsing context of the yara file object

5.171.2.28 deleteBindings()

```
void YaraFile::deleteBindings (
    YaraFileElement * element )
```

Removes all bindings in which participates given [YaraFileElement](#) object

5.171.2.29 dumpSyntaxErrors()

```
void YaraFile::dumpSyntaxErrors ( )
```

Prints formatted all syntax errors in the current file

5.171.2.30 edit() [1/2]

```
void YaraFile::edit (
    const offset_t & offset,
    const size_t & ins_n,
    const size_t & del_n )
```

Performs structural edit above this [YaraFile](#) (its tree structure). This method cannot be called twice without reparsing (that restore consistency of internal buffer).

Parameters

<i>offset</i>	The index of the byte where was file edited
<i>ins↔ _n</i>	Number of inserted characters by the edit
<i>del↔ _n</i>	Number of deleted characters by the edit

< Save the edit to the vector for reparsing

5.171.2.31 edit() [2/2]

```
void YaraFile::edit (
    const range_t & edited_range,
    const std::string & text )
```

Performs structural edit above this [YaraFile](#) (its tree structure). This method CAN be called multiple times without reparsing, because thanks to provided text it is possible to maintain internal buffer.

Parameters

<i>edited_range</i>	The index of the byte where was file edited
<i>text</i>	Inserted text

Note

Parameters have same meaning as parameters of `DidChangeTextDocument` in LSP

5.171.2.32 `getCommentByIndex()`

```
const std::unique_ptr< Comment > & YaraFile::getCommentByIndex (
    uint32_t index ) const
```

Returns local comment at specified index (the first comment that occurs has index 0)

5.171.2.33 `getComments()`

```
const SymTabOffset< Comment > & YaraFile::getComments ( ) const
```

Provides access to tab with comments

5.171.2.34 `getEdits()`

```
std::vector< edit_t > & YaraFile::getEdits ( )
```

Provides access to edit buffer

5.171.2.35 `getFileIncludes()`

```
std::vector< FileInclude * > YaraFile::getFileIncludes ( ) const
```

Returns all FileIncludes in file included included FileIncludes

5.171.2.36 `getImport()` [1/4]

```
Import * YaraFile::getImport (
    std::string_view name ) const
```

Searches for import that imports module with given name. Searching is done in this file and in included files.

Returns

Pointer to found module or nullptr if it was not found

< Return the first import that imports given module

5.171.2.37 getImport() [2/4]

```
Import * YaraFile::getImport (
    std::string_view name,
    const YaraFile * stop_file,
    std::vector< YaraFileElementBindable * > & includes )
```

Searches for [Import](#) object that imports module with given name until the stop file is found. Returns the dependent includes as the third argument. < If stop file is reached, interrupt searching

< Store dependent file includes or other [YaraFile](#) elements

5.171.2.38 getImport() [3/4]

```
Import * YaraFile::getImport (
    std::string_view name,
    offset_t before ) const
```

Searches for [Import](#) object that imports module with given name. [Import](#) must occurs before given offset. Searching is done in this file and in included files.

Returns

Pointer to found module or nullptr if it was not found

< Search for import in included file

< If there is import with given module name, check if any of them occurs before 'before' offset

5.171.2.39 getImport() [4/4]

```
Import * YaraFile::getImport (
    std::string_view name,
    offset_t before,
    std::vector< YaraFileElementBindable * > & includes,
    bool make_lookup_from_top = false ) const
```

Searches for [Import](#) object that imports module with given name. [Import](#) must occurs before given offset. Searching is done in this file and in included files and optionally in the whole parent [YaraSource](#). Returns the dependent includes as the third argument.

Returns

Pointer to found module or nullptr if it was not found

< Clear dependent includes

< Try to find import in included file

< Store dependent file includes or other [YaraFile](#) elements

< If there is no such import in included file, optionally try to find it from the start of parent [YaraSource](#)

5.171.2.40 getImports()

```
std::vector< Import * > YaraFile::getImports ( ) const
```

Returns the vector with pointers to all import statements, that can be accessed from this file (sorted by offset)

5.171.2.41 getLocalFileInclude()

```
FileInclude * YaraFile::getLocalFileInclude (
    uint32_t index ) const
```

Get local file includes by index

5.171.2.42 getLocalFileIncludes()

```
const SymTab< FileInclude > & YaraFile::getLocalFileIncludes ( ) const
```

Returns reference to symtab with file includes

5.171.2.43 getLocalImport()

```
Import * YaraFile::getLocalImport (
    uint32_t index ) const
```

Returns local [Import](#) object specified by the current index

5.171.2.44 getLocalImports()

```
const SymTabDuplId< Import > & YaraFile::getLocalImports ( ) const
```

Get map with imports in this file indexed by the offset of imports

5.171.2.45 getLocalRule()

```
const RulePtr & YaraFile::getLocalRule (
    const std::string & name ) const
```

Tries to find the rule by name in only in local context of this file (included files are not searched).

5.171.2.46 getLocalRuleByRow()

```
Rule * YaraFile::getLocalRuleByRow (
    uint32_t row )
```

Returns rule that overlaps given row, if there is no such rule, nullptr is returned

5.171.2.47 getLocalRules()

```
const SymTab< Rule > & YaraFile::getLocalRules ( ) const
```

Returns vector with the pointer to the rules in this [YaraFile](#). Vector is sorted by offset of the rules. If rule is added, removed or all rules are cleared returned, vector may not be valid. Caller must not delete any of rules from returned list.

5.171.2.48 getName()

```
const std::string & YaraFile::getName ( ) const
```

Returns the name of [YaraFile](#) (it is absolute path to file)

5.171.2.49 getRule() [1/3]

```
Rule * YaraFile::getRule (
    std::string_view name ) const
```

Returns pointer to rule with given name, that is located in this yara file or in any of included files

Returns

Pointer to the rule or nullptr (if it was not found)

5.171.2.50 getRule() [2/3]

```
Rule * YaraFile::getRule (
    std::string_view name,
    const YaraFile * stop_file,
    std::vector< YaraFileElementBindable * > & includes )
```

Searches for the rule with given name in this file and in included files until the stop_file is reached. Also returns the dependent file includes in third (output) parameter.

Returns

Pointer to the rule or nullptr

< Stop file is reached, interrupt searching

< Included file is not stop file, so try to find the rule

5.171.2.51 getRule() [3/3]

```
Rule * YaraFile::getRule (
    std::string_view name,
    offset_t before,
    std::vector< YaraFileElementBindable * > & includes,
    bool make_lookup_from_top = false ) const
```

Returns pointer to rule with given name, that is located in this yara file or in any of included files. Optionally the searching can be done from the start of [YaraSource](#). Also returns the dependent file includes in third (output) parameter (dependent file includes mean file includes that are necessary to have access to the rule in this yara_file).

Returns

Pointer to the rule or nullptr

< Try to find rule in included files

< Store dependent file include in the output vector

< [Rule](#) is located in the current yara file

5.171.2.52 getRules() [1/2]

```
std::vector< Rule * > YaraFile::getRules ( ) const
```

Get vector with pointers to the all rules located in this yara file and in included yara files

5.171.2.53 getRules() [2/2]

```
std::vector< Rule * > YaraFile::getRules (
    const std::string & name ) const
```

Get vector with pointers to the all rules located in this yara file and in included yara files, that have given name < Firstly, try to find rule name in current yara file

< Finally, try to find rule name in file includes

< Insert returned vector to the end of the result vector

5.171.2.54 getSemanticErrors()

```
const SymTabOffset< SemanticError > & YaraFile::getSemanticErrors ( ) const
```

Provides access to tab with syntax errors

5.171.2.55 getSyntaxErrors()

```
const SymTabOffset< SyntaxError > & YaraFile::getSyntaxErrors ( ) const
```

Provides access to tab with syntax errors

5.171.2.56 getTextFormatted()

```
std::stringstream YaraFile::getTextFormatted ( ) const [override], [virtual]
```

Provides serialization interface for objects that are inherited to [Printable](#)

Returns

Stringstream with serialized for of [Printable](#) object

Implements [Printable](#).

5.171.2.57 hasFastEdit()

```
bool YaraFile::hasFastEdit ( )
```

Check if there is fast edit (edit specified only by numerical values)

5.171.2.58 hasRule() [1/2]

```
bool YaraFile::hasRule (
    std::string_view name ) const
```

Check if the rule with given name is accessible in the whole context of this [YaraFile](#)

5.171.2.59 hasRule() [2/2]

```
bool YaraFile::hasRule (
    std::string_view name,
    offset_t before ) const
```

Check if the rule with given name is accessible in the context of this [YaraFile](#) before given offset < Check if rule has offset lower than given offset

5.171.2.60 isActive()

```
bool YaraFile::isActive ( )
```

Returns the state of activity flag

5.171.2.61 isIsolated()

```
bool YaraFile::isIsolated ( ) const
```

Checks if [YaraFile](#) object is isolated (isolated flag is set when include, that references this [YaraFile](#) object is deleted)

5.171.2.62 modify()

```
std::vector< offset_edit_range_t > YaraFile::modify (
    TSTree * new_tree )
```

Finds all offset ranges, that were modified by edits (important for deletes) or TS explicitly marks them as modified (it is important for contextual changes) < Traverse all performed edits

< Update ranges in edited_ranges vector (new edit can affect them)

< Push the new edit to the edited_ranges

< Merge ranges to avoid repeated construction of high level representation

5.171.2.63 mustBeReparsed()

```
bool YaraFile::mustBeReparsed ( )
```

Checks if there is any range, that should be checked (because semantic context has changed)

5.171.2.64 notifyAddedSymbol()

```
void YaraFile::notifyAddedSymbol (
    const std::string & id )
```

Checks missing symbols with specified id in the all included files

5.171.2.65 notifyChange()

```
void YaraFile::notifyChange (
    YaraFileElement * element )
```

Notifies change of [YaraFileElement](#) (dependencies will be afterwards also reparsed)

5.171.2.66 removeComments()

```
template<typename Fn >
uint32_t YaraFile::removeComments (
    const Fn & f )
```

Similar to [YaraFile::removeComments](#)

5.171.2.67 removeElementsInRange()

```
offset_range_t YaraFile::removeElementsInRange (
    const offset_t & start,
    const offset_t & end )
```

Removes all top level elements of [YaraFile](#) (and errors and comments) in specified range

5.171.2.68 removeGlobalErrors()

```
void YaraFile::removeGlobalErrors ( )
```

Removes all errors with global flag

5.171.2.69 removeLocalFileIncludes()

```
template<typename Fn >  
uint32_t YaraFile::removeLocalFileIncludes (   
    const Fn & f )
```

Removes all local file includes for which is evaluated given function as true

5.171.2.70 removeLocalImports()

```
template<typename Fn >  
uint32_t YaraFile::removeLocalImports (   
    const Fn & f )
```

Similar to [YaraFile::removeLocalRules](#)

5.171.2.71 removeLocalRules()

```
template<typename Fn >  
uint32_t YaraFile::removeLocalRules (   
    const Fn & f )
```

Removes the all rules, for that is evaluated given boolean function as true

5.171.2.72 removeMissingSymbol()

```
void YaraFile::removeMissingSymbol (   
    const std::string & missing )
```

Removes all missing symbols under with specified identifier

5.171.2.73 removeSemanticErrors()

```
template<typename Fn >  
uint32_t YaraFile::removeSemanticErrors (   
    const Fn & f )
```

Remove semantic errors specified by given lambda function (errors are deleted if lambda is evaluated for them as true)

5.171.2.74 removeSyntaxErrors()

```
template<typename Fn >
uint32_t YaraFile::removeSyntaxErrors (
    const Fn & f )
```

Remove syntax errors specified by given lambda function (errors are deleted if lambda is evaluated for them as true)

5.171.2.75 ruleLookup()

```
Rule * YaraFile::ruleLookup (
    const std::string & rule_id,
    std::vector< YaraFileElementBindable * > & includes )
```

Makes lookup in this file and in the parent [YaraSource](#) object

< Try to find same id rule in parent source

< Try To find local same id rule

5.171.2.76 setActivity()

```
void YaraFile::setActivity (
    bool state = true )
```

Sets the state of activity flag

5.171.2.77 setErrorMode()

```
void YaraFile::setErrorMode (
    YaramodConfig::ErrorMode new_mode )
```

Sets the error mode of the yara file object

Note

This function should be called by user, user should modify error mode by changing error mode of the parser

5.171.2.78 setIsolated()

```
void YaraFile::setIsolated (
    bool new_state = true )
```

Sets the sate of isolation flag

5.171.2.79 setName()

```
void YaraFile::setName (
    const std::string & new_name )
```

Sets the name of file

5.171.2.80 shiftElements()

```
void YaraFile::shiftElements (
    const offset_t & start,
    const uint32_t & start_row,
    const int32_t & delta,
    const int32_t & row_delta,
    const int32_t & col_delta )
```

Shifts all top level elements (their offset) of [YaraFile](#) by specified delta value (it can be also negative)

5.171.2.81 undo()

```
void YaraFile::undo ( )
```

Clears edit buffer and temporary string buffer

5.171.2.82 updateComments()

```
template<typename Fn >
void YaraFile::updateComments (
    const Fn & f )
```

Similar to [YaraFile::updateComments](#)

5.171.2.83 updateLocalFileIncludes()

```
template<typename Fn >
void YaraFile::updateLocalFileIncludes (
    const Fn & f )
```

Updates all local file includes by given lambda function

5.171.2.84 updateLocalImports()

```
template<typename Fn >
void YaraFile::updateLocalImports (
    const Fn & f )
```

Similar to [YaraFile::updateLocalRules](#)

5.171.2.85 updateLocalRules()

```
template<typename Fn >
void YaraFile::updateLocalRules (
    const Fn & f )
```

Updates local by given lambda function

5.171.2.86 updateSyntaxErrors()

```
template<typename Fn >
void YaraFile::updateSyntaxErrors (
    const Fn & f )
```

Updates all syntax error by given lambda function

5.171.2.87 wasEdited()

```
bool YaraFile::wasEdited ( )
```

Check if there is any pending edit in the edit buffer

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

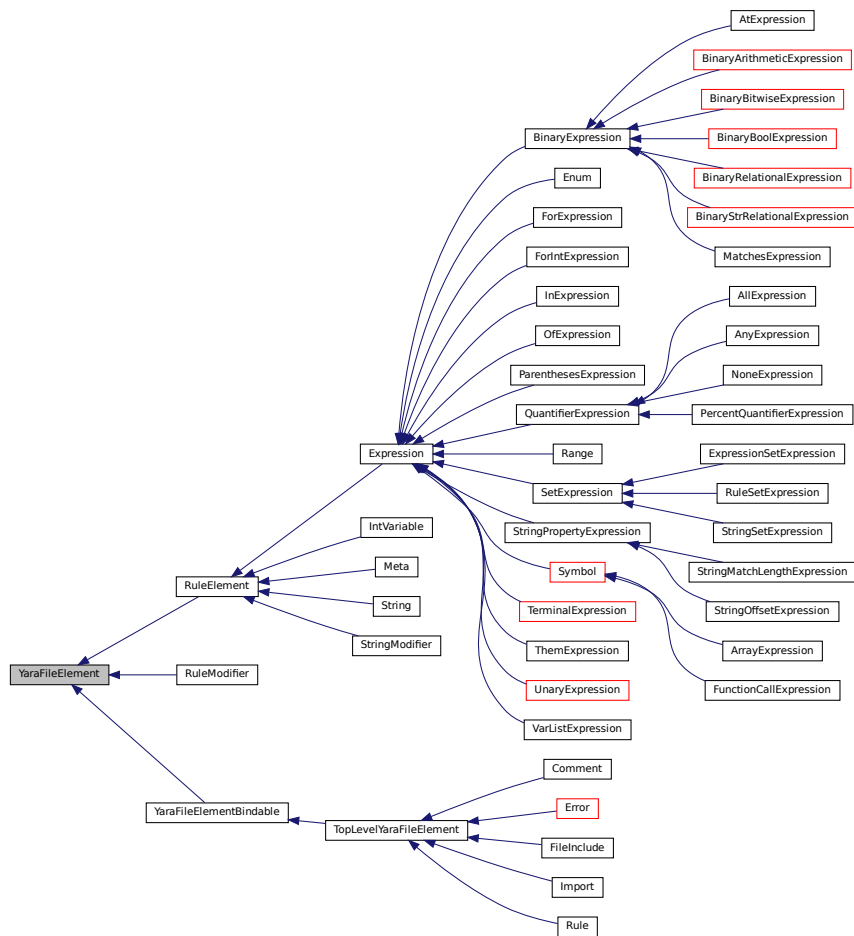
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file.cpp](#)

5.172 YaraFileElement Class Reference

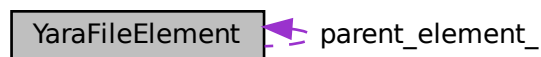
Class representing syntax structures of [YaraFile](#) (such as Rules, Imports, Includes - everything which has some characters in the source string)

```
#include <yara_file_element.h>
```

Inheritance diagram for YaraFileElement:



Collaboration diagram for YaraFileElement:



Public Member Functions

- **YaraFileElement** (const offset_t &offset, const size_t &len)
- virtual offset_t [getOffset](#) () const
- virtual void [setOffset](#) (const offset_t &offset)
- size_t [getLen](#) () const
- void [setLen](#) (const size_t &len)
- [point_t](#) [getPosition](#) () const

- [range_t](#) `getRange ()` const
- void `setParentFile` (const std::shared_ptr< [YaraFile](#) > &parent)
- std::weak_ptr< [YaraFile](#) > `getParentFile ()` const
- bool `isDeletionRequested ()` const
- void `requestDeletion ()`
- bool `isFixed ()` const
- void `setParent` ([YaraFileElement](#) *parent)

Protected Attributes

- offset_t `offset_`
- size_t `len_`
- bool `is_deletion_requested_` = false
- std::weak_ptr< [YaraFile](#) > `parent_file_`
- [YaraFileElement](#) * `parent_element_` = nullptr

5.172.1 Detailed Description

Class representing syntax structures of [YaraFile](#) (such as Rules, Imports, Includes - everything which has some characters in the source string)

5.172.2 Member Function Documentation

5.172.2.1 `getLen()`

```
size_t YaraFileElement::getLen ( ) const
```

Returns length of yara file element

5.172.2.2 `getOffset()`

```
offset_t YaraFileElement::getOffset ( ) const [virtual]
```

Returns start offset of yara file element

Reimplemented in [RuleElement](#).

5.172.2.3 `getParentFile()`

```
std::weak_ptr< YaraFile > YaraFileElement::getParentFile ( ) const
```

Gets weak_ptr to parent file

5.172.2.4 getPosition()

```
point_t YaraFileElement::getPosition ( ) const
```

Returns position of first character of the yara file element

5.172.2.5 getRange()

```
range_t YaraFileElement::getRange ( ) const
```

Returns [Range](#) where is [YaraFileElement](#) located

5.172.2.6 isDeletionRequested()

```
bool YaraFileElement::isDeletionRequested ( ) const
```

Checks if deletion of this [YaraFileElement](#) was requested

5.172.2.7 isFixed()

```
bool YaraFileElement::isFixed ( ) const
```

Check if [YaraFileElement](#) is fixed to other [YaraFileElement](#)

5.172.2.8 requestDeletion()

```
void YaraFileElement::requestDeletion ( )
```

Request deletion of this [YaraFileElement](#)

5.172.2.9 setLen()

```
void YaraFileElement::setLen (
    const size_t & len )
```

Sets the length of yara file element

5.172.2.10 setOffset()

```
void YaraFileElement::setOffset (
    const offset_t & offset ) [virtual]
```

Sets the start offset of yara file element

Reimplemented in [RuleElement](#).

5.172.2.11 setParent()

```
void YaraFileElement::setParent (
    YaraFileElement * parent )
```

Sets [YaraFileElement](#) to this [YaraFileElement](#) as parent

5.172.2.12 setParentFile()

```
void YaraFileElement::setParentFile (
    const std::shared_ptr< YaraFile > & parent )
```

Sets parent file of element

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

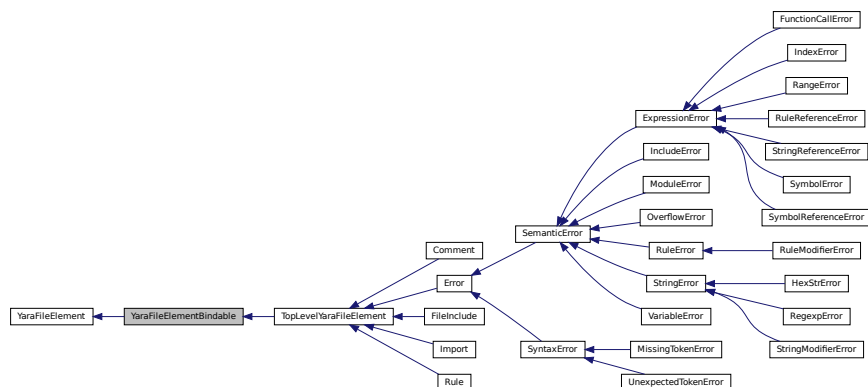
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file_element.h](#)
- [/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file_element.cpp](#)

5.173 YaraFileElementBindable Class Reference

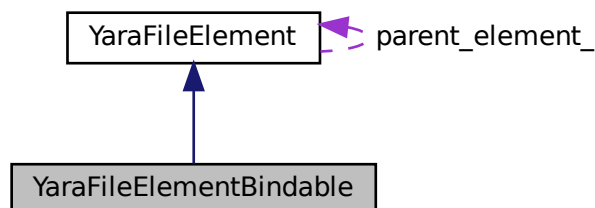
Represents of elements of [YaraFile](#), that can have some semantic binding with other elements.

```
#include <yara_file_element.h>
```

Inheritance diagram for YaraFileElementBindable:



Collaboration diagram for YaraFileElementBindable:



Public Member Functions

- **YaraFileElementBindable** (const offset_t &offset, const size_t &len)
- **YaraFileElementBindable** * **bind** (**YaraFileElementBindable** *dependency)
- **YaraFileElementBindable** * **bind_to** (**YaraFileElementBindable** *parent)

Additional Inherited Members

5.173.1 Detailed Description

Represents of elements of [YaraFile](#), that can have some semantic binding with other elements.

5.173.2 Member Function Documentation

5.173.2.1 bind()

```
YaraFileElementBindable * YaraFileElementBindable::bind (  
    YaraFileElementBindable * dependency )
```

Creates semantic binding between "this" and other element, that is given as argument. This element is

5.173.2.2 bind_to()

```
YaraFileElementBindable * YaraFileElementBindable::bind_to (  
    YaraFileElementBindable * parent )
```

Creates semantic binding between "this" and other element

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[yara_file_element.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/[yara_file_element.cpp](#)

5.174 Yaramod Class Reference

The class, that controls parsing of Yara rulesets.

```
#include <yaramod.h>
```

Public Member Functions

- **Yaramod** (const std::string &base_dir, const std::string &module_dir="modules", YaramodConfig::ParsingMode p_mode=YaramodConfig::ParsingMode::Auto, YaramodConfig::ErrorMode e_mode=YaramodConfig::ErrorMode::Tolerant) const
- YaraSourcePtr **parseFile** (std::string_view path) const
- void **parseFile** (std::string_view path, const YaraSourcePtr &old_src) const
- void **parseFile** (std::string_view path, YaraSource *old_src) const
- YaraSourcePtr **parseString** (const std::string &string) const
- YaraSourcePtr **parseString** (const std::string &string, const std::string &entry_file_name) const
- void **parseString** (const std::string &string, const YaraSourcePtr &old_src) const
- void **parseString** (const std::string &string, YaraSource *old_src) const
- void **parseString** (const std::string &string, const YaraSourcePtr &old_src, const std::string &entry_file_name) const
- void **parseString** (const std::string &string, YaraSource *old_src, const std::string &entry_file_name) const
- YaramodConfig & **config** ()
- void **loadModulesFromPath** (std::string_view path)
- void **loadModulesFromEnvironment** ()
- void **loadModules** ()

5.174.1 Detailed Description

The class, that controls parsing of Yara rulesets.

The instance of this class is created by the user and user is also responsible for deleting of it.

5.174.2 Member Function Documentation

5.174.2.1 config()

```
YaramodConfig & Yaramod::config ( )
```

Provides access to configuration object

5.174.2.2 loadModules()

```
void Yaramod::loadModules ( )
```

Loads (parses) the JSONs with module descriptions

5.174.2.3 loadModulesFromEnvironment()

```
void Yaramod::loadModulesFromEnvironment ( )
```

Loads modules from paths, that are in MODULE_PATH_ENV_VAR

5.174.2.4 loadModulesFromPath()

```
void Yaramod::loadModulesFromPath (
    std::string_view path )
```

Load module(s) located at given path

Note

if path represents directory, it tries to load all modules in it, if path is regular file, it is considered as standalone module

5.174.2.5 parseFile() [1/2]

```
YaraSourcePtr Yaramod::parseFile (
    std::string_view path ) const
```

Parses yara file given by path (and included files)

Parameters

<i>path</i>	Path of the entry file
-------------	------------------------

Returns

Pointer to [YaraSource](#) object

5.174.2.6 parseFile() [2/2]

```
void Yaramod::parseFile (
    std::string_view path,
    const YaraSourcePtr & old_src ) const
```

Updates [YaraSource](#) object after edits (edit method must be called to perform structural edit first -

See also

[YaraSource](#)).

Parameters

<i>path</i>	Path of the entry file (must be the same as was when was YaraSource created)
<i>old_src</i>	Instance of YaraSource should be updated

Returns

Reference to pointer to updated [YaraSource](#) (only to be consistent with [Yaramod::parseFile](#))

5.174.2.7 parseString() [1/2]

```
YaraSourcePtr Yaramod::parseString (
    const std::string & string ) const
```

Parses yara file given by C++ string. Files, that are included in this string are also parsed.

Parameters

<i>string</i>	The content of the entry file
---------------	-------------------------------

Returns

Pointer to [YaraSource](#) object

5.174.2.8 parseString() [2/2]

```
void Yaramod::parseString (
    const std::string & string,
    const YaraSourcePtr & old_src ) const
```

Updates [YaraSource](#) object after edits (edit method must be called to perform structural edit first -

See also

[YaraSource](#)).

Parameters

<i>string</i>	Content of the entry file (edited or original)
<i>old_src</i>	Instance of YaraSource should be updated

Returns

Reference to pointer to updated [YaraSource](#) (only to be consistent with [Yaramod::parseFile](#))

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[yaramod.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/[yaramod.cpp](#)

5.175 YaramodConfig Class Reference

Objects of this class hold the user defined configuration of parser, that can be changed in runtime.

```
#include <yaramod_config.h>
```

Public Types

- enum class [ErrorMode](#) { [Tolerant](#) , [Strict](#) }
The modes of error collectors.
- enum class [ParsingMode](#) { [Auto](#) , [Manual](#) }
The modes of parser.

Public Member Functions

- [YaramodConfig](#) ()=default
- [YaramodConfig](#) (const [YaramodConfig](#) &)=delete
- [YaramodConfig](#) & **operator=** (const [YaramodConfig](#) &)=delete
- void [setErrorMode](#) ([ErrorMode](#) mode)
- [ErrorMode](#) [getErrorMode](#) ()
- void [setParsingMode](#) ([ParsingMode](#) mode)
- [ParsingMode](#) [getParsingMode](#) ()
- void [setBaseDir](#) (const std::string &path)
- const std::string & [getBaseDir](#) ()
- void [setModuleDir](#) (const std::string &path)
- const std::string & [getModuleDir](#) ()
- void [addVar](#) (const std::string &var_name, int64_t value)
- void [addVar](#) (const std::string &var_name, const std::string &value)
- void [addBoolVar](#) (const std::string &var_name, bool value)
- void [removeVar](#) (const std::string &var_name)
- [ExtVariable](#) * [getVar](#) (const std::string &var_name)

5.175.1 Detailed Description

Objects of this class hold the user defined configuration of parser, that can be changed in runtime.

5.175.2 Member Enumeration Documentation

5.175.2.1 ErrorMode

```
enum YaramodConfig::ErrorMode [strong]
```

The modes of error collectors.

Enumerator

Tolerant	Errors are collected to map (default)
Strict	When error is gouging to be added to map exception is thrown.

5.175.2.2 ParsingMode

```
enum YaramodConfig::ParsingMode [strong]
```

The modes of parser.

Enumerator

Auto	Included files are parsed automatically (default)
Manual	Parsing of included files must be requested manually.

5.175.3 Constructor & Destructor Documentation**5.175.3.1 YaramodConfig()**

```
YaramodConfig::YaramodConfig ( ) [default]
```

Class is container of unique_ptr, so copy ctor, assignment operator must be explicitly deleted (otherwise it causes problems with pybind11)

5.175.4 Member Function Documentation**5.175.4.1 addBoolVar()**

```
void YaramodConfig::addBoolVar (
    const std::string & var_name,
    bool value )
```

Adds boolean variable to variables

Parameters

<i>var_name</i>	The name of variable
<i>value</i>	Value stored in the variable

Note

It has the same functionality as [YaramodConfig::addVar](#), but this is not another overload, because it may conflict with other var data types, that can be evaluated as bool

5.175.4.2 addVar()

```
void YaramodConfig::addVar (
    const std::string & var_name,
    int64_t value )
```

Adds external variable to the database, if variable with the same name already exists, it is rewritten by the new value

Parameters

<i>var_name</i>	The name of variable
<i>value</i>	Value stored in the variable

5.175.4.3 getBaseDir()

```
const std::string & YaramodConfig::getBaseDir ( )
```

Gets the path of the base directory

5.175.4.4 getErrorMode()

```
YaramodConfig::ErrorMode YaramodConfig::getErrorMode ( )
```

Gets the mode of error collectors

5.175.4.5 getModuleDir()

```
const std::string & YaramodConfig::getModuleDir ( )
```

Gets the path of directory with modules

5.175.4.6 getParsingMode()

```
YaramodConfig::ParsingMode YaramodConfig::getParsingMode ( )
```

Gets the mode of parsing includes

5.175.4.7 getVar()

```
ExtVariable * YaramodConfig::getVar (
    const std::string & var_name )
```

Provides access to variable with specified name

Parameters

<code>var_name</code>	Name of variable, that will be returned
-----------------------	---

Returns

Pointer to [ExtVariable](#) object with given name or nullptr

5.175.4.8 removeVar()

```
void YaramodConfig::removeVar (
    const std::string & var_name )
```

Removes variable from database of external variables

Note

If variable does not exists, nothing happens

Parameters

<code>var_name</code>	Name of variable, that will be removed
-----------------------	--

5.175.4.9 setBaseDir()

```
void YaramodConfig::setBaseDir (
    const std::string & path )
```

Sets the path of the base directory (path of directory for every relative path that is used in include statements)

5.175.4.10 setErrorMode()

```
void YaramodConfig::setErrorMode (
    ErrorMode mode )
```

Sets the mode of error collectors

5.175.4.11 setModuleDir()

```
void YaramodConfig::setModuleDir (
    const std::string & path )
```

Sets the path of directory with modules

Note

In this directory should be located module descriptions in JSON format. After changing of module directory, [Yaramod::loadModules](#) should be called to load JSONs located in the new directory.

5.175.4.12 setParsingMode()

```
void YaramodConfig::setParsingMode (
    ParsingMode mode )
```

Sets the mode of parsing includes

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

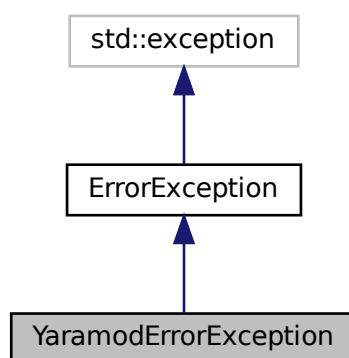
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[yaramod_config.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/[yaramod_config.cpp](#)

5.176 YaramodErrorException Class Reference

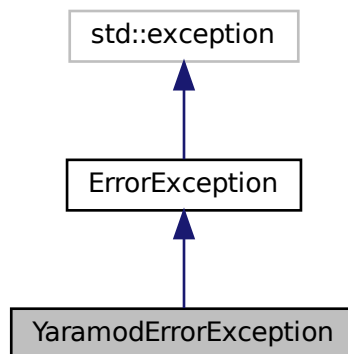
Exception for other errors, that can occur during parsing.

```
#include <error.h>
```

Inheritance diagram for YaramodErrorException:



Collaboration diagram for YaramodErrorException:



Public Member Functions

- **YaramodErrorException** (std::string msg={})

5.176.1 Detailed Description

Exception for other errors, that can occur during parsing.

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

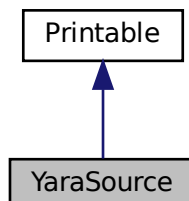
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[error.h](#)

5.177 YaraSource Class Reference

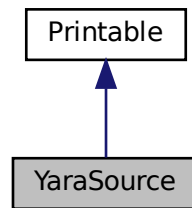
Class representing the set of files that are connected by includes.

```
#include <yara_source.h>
```

Inheritance diagram for YaraSource:



Collaboration diagram for YaraSource:



Public Types

- using **FileMap** = std::unordered_map< std::string, std::shared_ptr< [YaraFile](#) >, [string_hash_t](#), std::equal<_to<> >

Public Member Functions

- const std::shared_ptr< [YaraFile](#) > & [getEntryFile](#) () const
- std::shared_ptr< [YaraFile](#) > [getFile](#) (std::string_view path) const
- bool [hasFile](#) (std::string_view path)
- void [addFile](#) (const std::shared_ptr< [YaraFile](#) > &file_ptr)
- void [setEntryFile](#) (std::string_view path)
- void [removeFile](#) (const std::string &path)
- bool [renameFile](#) (std::string_view old_path, std::string_view new_path)
- const FileMap & [getFiles](#) () const
- void [clearInactiveFiles](#) ()
- void [deactivateFiles](#) ()
- void [edit](#) (const offset_t &offset, const size_t &ins_n, const size_t &del_n)
- void [edit](#) (const offset_t &offset, const size_t &ins_n, const size_t &del_n, const std::string &path)
- void [edit](#) (const [range_t](#) &edited_range, const std::string &text)
- void [edit](#) (const [range_t](#) &edited_range, const std::string &text, const std::string &path)
- void [undo](#) ()
- std::stringstream [getTextFormatted](#) () const
- const std::string & [getBaseDir](#) ()
- void [setBaseDir](#) (std::string_view path)
- void [notifyChange](#) ([YaraFileElement](#) *element)
- void [bind](#) ([YaraFileElement](#) *parent, [YaraFileElement](#) *dependency)
- void [deleteBindings](#) ([YaraFileElement](#) *element)
- void [moveBindings](#) ([YaraFileElement](#) *original_parent, [YaraFileElementBindable](#) *new_parent)
- void [dumpBindChildren](#) ()
- void [dumpBindParents](#) ()
- std::vector< [Rule](#) * > [getAllRules](#) ()
- std::vector< [SyntaxError](#) * > [getAllSyntaxErrors](#) ()
- std::vector< [SemanticError](#) * > [getAllSemanticErrors](#) ()
- [Import](#) * [getImport](#) (std::string_view name, const [YaraFile](#) *stop_file, std::vector< [YaraFileElementBindable](#) * > &includes) const
- [Rule](#) * [getRule](#) (std::string_view name, const [YaraFile](#) *stop_file, std::vector< [YaraFileElementBindable](#) * > &includes) const
- bool [hasPendingChanges](#) () const

5.177.1 Detailed Description

Class representing the set of files that are connected by includes.

There is one entry file with N included files and in included files there can be other includes.

5.177.2 Member Function Documentation

5.177.2.1 `addFile()`

```
void YaraSource::addFile (
    const std::shared_ptr< YaraFile > & file_ptr )
```

Adds [YaraFile](#) object to [YaraSource](#)

5.177.2.2 `bind()`

```
void YaraSource::bind (
    YaraFileElement * parent,
    YaraFileElement * dependency )
```

Creates semantic binding between two yara file elements < Creates bind between parent and dependency

< Create reversed binding to easily delete forward binding

5.177.2.3 `clearInactiveFiles()`

```
void YaraSource::clearInactiveFiles ( )
```

Removes all files, that have not set activity flag < Clear all files that have not set activity flag (they are not included in yara source)

5.177.2.4 `deactivateFiles()`

```
void YaraSource::deactivateFiles ( )
```

Unset the activity flag of all the files stored in [YaraSource](#)

5.177.2.5 `deleteBindings()`

```
void YaraSource::deleteBindings (
    YaraFileElement * element )
```

Delete all bindings in which participate given element < Destroy all bindings where element acts as child

< Destroy all bindings where element acts as child

< Destroy all bindings where element acts as parent

< Remove reverse binding of element

5.177.2.6 dumpBindChildren()

```
void YaraSource::dumpBindChildren ( )
```

Prints ranges of bindings from point of view parents

5.177.2.7 dumpBindParents()

```
void YaraSource::dumpBindParents ( )
```

Prints ranges of bindings from point of view children

5.177.2.8 edit() [1/4]

```
void YaraSource::edit (
    const offset_t & offset,
    const size_t & ins_n,
    const size_t & del_n )
```

Reports an edit of entry file. Edits must be reported before reparsing for correct incremental parsing process.

Parameters

<i>offset</i>	Offset where edit starts (offset of first changed character)
<i>ins</i> _↔ <i>_n</i>	Number of inserted characters
<i>del</i> _↔ <i>_n</i>	Number of deleted characters

Warning

After use of this overload of edit function, reparsing must be done, because there is now way how to update the content of internal [YaraFile](#) buffer! (because the inserted text is not provided)

5.177.2.9 edit() [2/4]

```
void YaraSource::edit (
    const offset_t & offset,
    const size_t & ins_n,
    const size_t & del_n,
    const std::string & path )
```

Reports an edit of file with given path. Edits must be reported before reparsing for correct incremental parsing process.

Parameters

<i>offset</i>	Offset where edit starts (offset of first changed character)
<i>ins↔ _n</i>	Number of inserted characters
<i>del↔ _n</i>	Number of deleted characters
<i>path</i>	Path of edited yara file object

Warning

After use of this overload of edit function, reparsing must be done, because there is now way how to update the content of internal [YaraFile](#) buffer! (because the inserted text is not provided)

5.177.2.10 `edit()` [3/4]

```
void YaraSource::edit (
    const range\_t & edited_range,
    const std::string & text )
```

Reports an edit of entry file. Edits must be reported before reparsing for correct incremental parsing process. It should correspond to the way how are the edits reported in LSP.

Parameters

<i>edited_range</i>	Range where edit was done (start pt is location of the first edited character and end pt is location of the first character that is not affected by the edit)
<i>text</i>	Newly inserted text (if only deletion was done use empty string here)

Note

By this overload of edit method, multiple edits can be reported before reparsing.

5.177.2.11 `edit()` [4/4]

```
void YaraSource::edit (
    const range\_t & edited_range,
    const std::string & text,
    const std::string & path )
```

Reports an edit of file with given name. Edits must be reported before reparsing for correct incremental parsing process. It should correspond to the way how are the edits reported in LSP.

Parameters

<i>edited_range</i>	Range where edit was done (start pt is location of the first edited character and end pt is location of the first character that is not affected by the edit)
<i>text</i>	Newly inserted text (if only deletion was done use empty string here)

Note

By this overload of edit method, multiple edits can be reported before reparsing.

5.177.2.12 getAllRules()

```
std::vector< Rule * > YaraSource::getAllRules ( )
```

Returns vector with pointers to all rules included in [YaraSource](#) object

Note

This way is not effective (the new vector is allocated and files syntax rules are sequentially copied), so use it only if worse performance does not matter

5.177.2.13 getAllSemanticErrors()

```
std::vector< SemanticError * > YaraSource::getAllSemanticErrors ( )
```

Returns vector with pointers to all semantic errors located in [YaraSource](#) object

Note

This way is not effective (the new vector is allocated and files syntax rules are sequentially copied), so use it only if worse performance does not matter

5.177.2.14 getAllSyntaxErrors()

```
std::vector< SyntaxError * > YaraSource::getAllSyntaxErrors ( )
```

Returns vector with pointers to all syntax errors located in [YaraSource](#) object

Note

This way is not effective (the new vector is allocated and files syntax rules are sequentially copied), so use it only if worse performance does not matter

5.177.2.15 `getBaseDir()`

```
const std::string & YaraSource::getBaseDir ( )
```

Returns base directory path of yara source (under normal circumstances it is path to directory with the entry file)

5.177.2.16 `getEntryFile()`

```
const std::shared_ptr< YaraFile > & YaraSource::getEntryFile ( ) const
```

Returns pointer to entry [YaraFile](#) object

5.177.2.17 `getFile()`

```
std::shared_ptr< YaraFile > YaraSource::getFile (
    std::string_view path ) const
```

Returns pointer to [YaraFile](#) object with given path

Note

if [YaraFile](#) with provided path does not exists exception is thrown

5.177.2.18 `getFiles()`

```
const YaraSource::FileMap & YaraSource::getFiles ( ) const
```

Returns container with all files stored in [YaraSource](#)

5.177.2.19 `getImport()`

```
Import * YaraSource::getImport (
    std::string_view name,
    const YaraFile * stop_file,
    std::vector< YaraFileElementBindable * > & includes ) const
```

Find the first occurrence of import, that imports module with given name. If stop_file is reached, searching is interrupted.

Returns

Pointer to [Import](#) object or nullptr

5.177.2.20 getRule()

```
Rule * YaraSource::getRule (
    std::string_view name,
    const YaraFile * stop_file,
    std::vector< YaraFileElementBindable * > & includes ) const
```

Find the first occurrence of rule with given id. If stop_file is, reached searching is interrupted.

Returns

Pointer to [Import](#) object or nullptr

5.177.2.21 getTextFormatted()

```
std::stringstream YaraSource::getTextFormatted ( ) const [virtual]
```

Returns stringstream with formatted content of all yara files

Note

The order of yara files is preserved

Implements [Printable](#).

5.177.2.22 hasFile()

```
bool YaraSource::hasFile (
    std::string_view path )
```

Check if there is [YaraFile](#) object in [YaraSource](#)

Returns

true if there is, otherwise false

5.177.2.23 moveBindings()

```
void YaraSource::moveBindings (
    YaraFileElement * original_parent,
    YaraFileElementBindable * new_parent )
```

Switch bindings of original parent to new parent element

5.177.2.24 notifyChange()

```
void YaraSource::notifyChange (
    YaraFileElement * element )
```

Notifies change of yara file element (child elements can be afterwards checked) < Iterate over all children of element

< Get their range

< If dependent element has fixed position that depends on this element, request its deletion

< Add range to check_ranges (these ranges will be reparsed)

5.177.2.25 removeFile()

```
void YaraSource::removeFile (
    const std::string & path )
```

Removes [YaraFile](#) object with provided path from [YaraSource](#)

5.177.2.26 renameFile()

```
bool YaraSource::renameFile (
    std::string_view old_path,
    std::string_view new_path )
```

Renames file in [YaraSource](#)

5.177.2.27 setEntryFile()

```
void YaraSource::setEntryFile (
    std::string_view path )
```

Sets entry file of [YaraSource](#)

Note

[YaraFile](#) object must be first added by [YaraSource::addFile](#) method

< File must be already registered in this [YaraSource](#)

5.177.2.28 undo()

```
void YaraSource::undo ( )
```

Remove all changes, that were reported by the [YaraSource::edit](#) and [YaraFile::edit](#) calls since the last (re)parsing in the whole [YaraSource](#) object

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

- /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/[yara_source.h](#)
- /home/vojtech.dvorak1/Documents/yaramod-v4/src/[yara_source.cpp](#)

Chapter 6

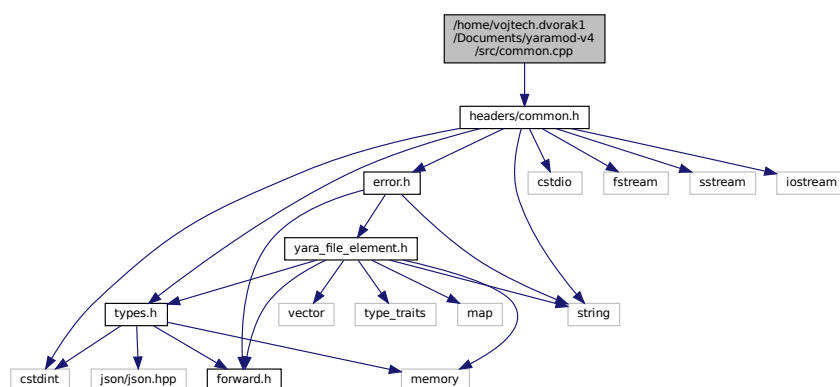
File Documentation

6.1 /home/vojtech.dvorak1/Documents/yaramod-v4/src/common.cpp File Reference

Implementation of auxiliary functions and methods, that are used across the project.

```
#include "headers/common.h"
```

Include dependency graph for common.cpp:



Functions

- `std::string readFileToString (std::string_view path)`
Reads file given by path to and assigns its content to given string.
- `offset_t pointToOffset (std::string_view str, const point_t &point, const point_t &start_pt, const offset_t &start_offset)`
Converts point to offset, this conversion depends on string str which must be provided as the first argument of this function.
- `point_t offsetToPoint (std::string_view str, const offset_t &offset, const point_t &start_pt, const offset_t &start_offset)`
Converts offset to point (string must be provided as the first argument)
- `std::string normalizeFSPath (const std::string &path_str)`
Creates normalized form of given path and returns it as a string.

6.1.1 Detailed Description

Implementation of auxiliary functions and methods, that are used across the project.

Author

Vojtěch Dvořák

6.1.2 Function Documentation

6.1.2.1 normalizeFSPath()

```
std::string normalizeFSPath (
    const std::string & path_str )
```

Creates normalized form of given path and returns it as a string.

Note

Just normalizes path, check if path exists must be done by caller

6.1.2.2 offsetToPoint()

```
point_t offsetToPoint (
    std::string_view str,
    const offset_t & offset,
    const point_t & start_pt,
    const offset_t & start_offset )
```

Converts offset to point (string must be provided as the first argument)

< End of the line reached

6.1.2.3 pointToOffset()

```
offset_t pointToOffset (
    std::string_view str,
    const point_t & point,
    const point_t & start_pt,
    const offset_t & start_offset )
```

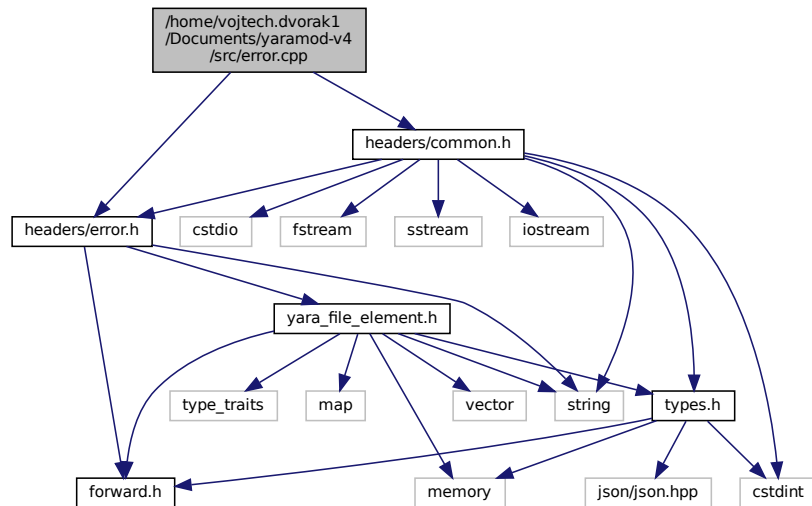
Converts point to offset, this conversion depends on string str which must be provided as the first argument of this function.

< End of the line reached

6.2 /home/vojtech.dvorak1/Documents/yaramod-v4/src/error.cpp File Reference

Implementation of members of error and exception classes.

```
#include "headers/error.h"
#include "headers/common.h"
Include dependency graph for error.cpp:
```



6.2.1 Detailed Description

Implementation of members of error and exception classes.

Author

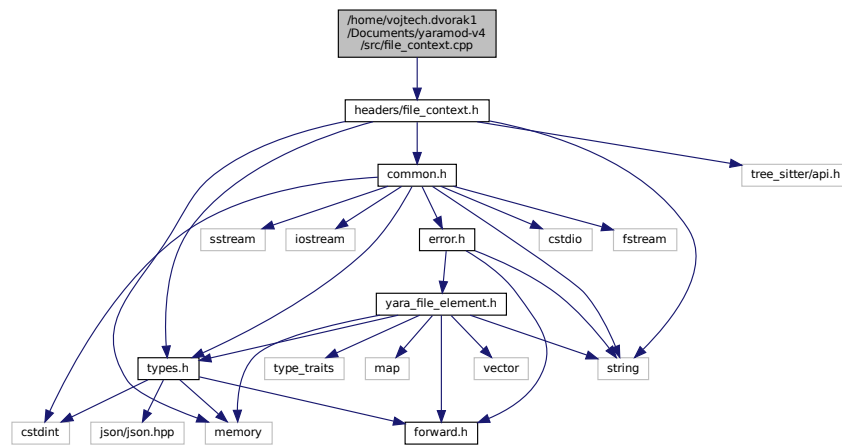
Vojtěch Dvořák

6.3 /home/vojtech.dvorak1/Documents/yaramod-v4/src/file_context.cpp File Reference

Contains implementation of [FileContext](#) methods.

```
#include "headers/file_context.h"
```

Include dependency graph for file_context.cpp:



6.3.1 Detailed Description

Contains implementation of [FileContext](#) methods.

Author

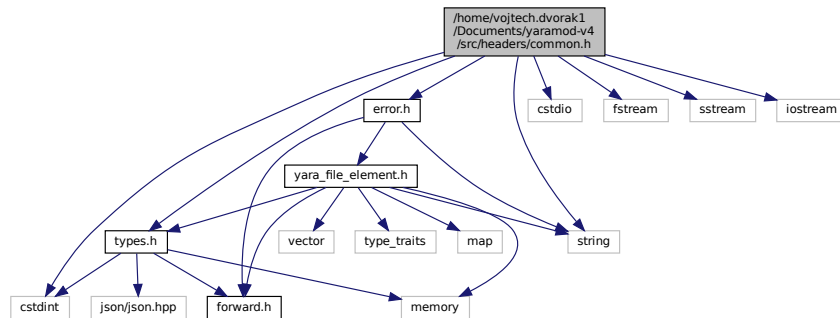
Vojtěch Dvořák

6.4 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/common.h File Reference

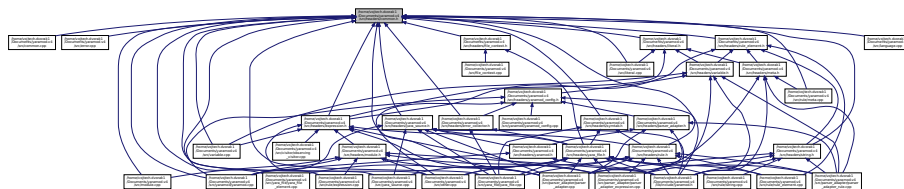
Auxiliary functions and classes, that can be used across the project.

```
#include "types.h"
#include "error.h"
#include <string>
#include <cstdint>
#include <fstream>
#include <sstream>
#include <iostream>
#include <cstdint>
```

Include dependency graph for common.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Printable](#)
Base class for types, that can be serialized.

Macros

- `#define DEBUG_LOG(...)`
- `#define TO_STR(NUM) #NUM`
- `#define TO_STR(NUM) TO_STR(NUM)`

Functions

- `std::string readFileToString (std::string_view path)`
Reads file given by path to and assigns its content to given string.
- `offset_t pointToOffset (std::string_view str, const point_t &point, const point_t &start_pt={0, 0}, const offset_t &start_offset=0)`
Converts point to offset, this conversion depends on string str which must be provided as the first argument of this function.
- `point_t offsetToPoint (std::string_view str, const offset_t &offset, const point_t &start_pt={0, 0}, const offset_t &start_offset=0)`
Converts offset to point (string must be provided as the first argument)
- `std::string normalizeFSPath (const std::string &path_str)`
Creates normalized form of given path and returns it as a string.

6.4.1 Detailed Description

Auxiliary functions and classes, that can be used across the project.

Author

Vojtěch Dvořák

6.4.2 Function Documentation

6.4.2.1 normalizeFSPath()

```
std::string normalizeFSPath (
    const std::string & path_str )
```

Creates normalized form of given path and returns it as a string.

Note

Just normalizes path, check if pat exists must be done by caller

6.4.2.2 offsetToPoint()

```
point_t offsetToPoint (
    std::string_view str,
    const offset_t & offset,
    const point_t & start_pt = {0, 0},
    const offset_t & start_offset = 0 )
```

Converts offset to point (string must be provided as the first argument)

< End of the line reached

6.4.2.3 pointToOffset()

```
offset_t pointToOffset (
    std::string_view str,
    const point_t & point,
    const point_t & start_pt = {0, 0},
    const offset_t & start_offset = 0 )
```

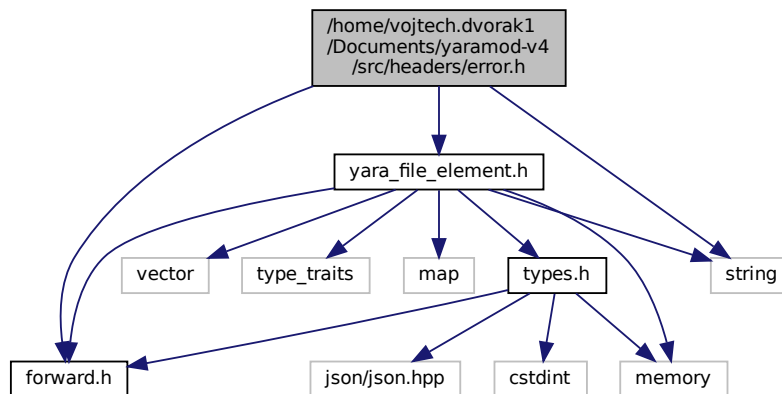
Converts point to offset, this conversion depends on string str which must be provided as the first argument of this function.

< End of the line reached

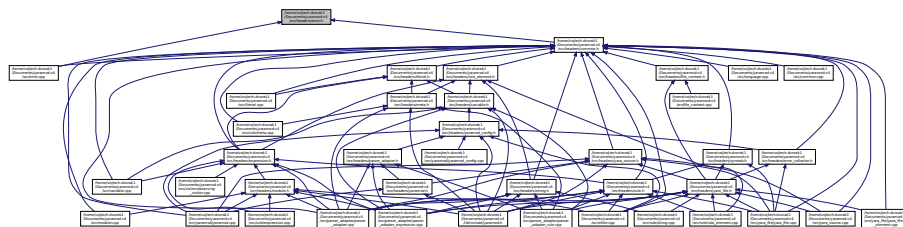
6.5 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h File Reference

Contains declaration of error and exception hierarchies.

```
#include "yara_file_element.h"
#include <string>
#include "forward.h"
Include dependency graph for error.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [ErrorException](#)
Base class for exceptions.
- class [InternalErrorException](#)
Exception for unexpected errors, that are probably not caused by user (nonrecoverable errors of internal parser etc.)
- class [YaramodErrorException](#)
Exception for other errors, that can occur during parsing.
- class [YaraErrorException](#)
Base Exception class for errors located in input YARA code.
- class [SyntaxErrorException](#)
Exception representing syntax error in parsed source code.

- class [UnexpectedTokenErrorException](#)
Exception representing syntax error in parsed source code.
- class [MissingTokenErrorException](#)
Exception representing missing token in parsed source code.
- class [SemanticErrorException](#)
Exception representing semantic errors in parsed source code.
- class [ExpressionErrorException](#)
Base class for exception, that represents error, that occurred in expression
- class [IndexErrorException](#)
Exception for bad type of index expression in array access expression
- class [RuleReferenceErrorException](#)
Exception for bad rule reference (e. g. rule does not exists)
- class [StringReferenceErrorException](#)
Exception for bad string reference (e. g. string does not exists)
- class [RangeErrorException](#)
Exception for errors, that occur in range expression (e. g. higher left bound than right bound)
- class [FunctionCallErrorException](#)
Exception for errors, that occur in function call (e. g. bad types of arguments)
- class [SymbolReferenceErrorException](#)
Exception for bad symbol reference (e. g. symbol does not exists)
- class [SymbolErrorException](#)
Exception for bad usage of symbol (e. g. symbol is used as structure, but it is value)
- class [IncludeErrorException](#)
Exception for bad file include (e. g. file is included twice)
- class [StringErrorException](#)
Exception for errors, that occurred in string definition (e. g. collision of string identifier)
- class [StringModifierErrorException](#)
Exception error of string modifier (e. g. invalid combination of string type and modifier)
- class [RegexpErrorException](#)
Exception of semantic error of regular expression (e. g. bad repeat interval)
- class [HexStrErrorException](#)
Exception for semantic errors of hexadecimal string (e. g. bad jump interval)
- class [OverflowErrorException](#)
Exception for overflows.
- class [ModuleErrorException](#)
Exception for errors, that occurred in module import
- class [RuleErrorException](#)
Exception for errors occurred in rule definition (e. g. collision of rule identifier)
- class [RuleModifierErrorException](#)
Exception for errors of rule modifier (e. g. duplicated rule modifier)
- class [VariableErrorException](#)
Exception for errors in variable definition.
- class [Error](#)
Class, that represents errors in analyzed source code. The main difference between ErrorExceptions and Errors is in usage - Errors are stored in some vector/list or in something and [ErrorException](#) are thrown.
- class [SyntaxError](#)
Base class for syntax errors.
- class [UnexpectedTokenError](#)

- Class for syntax error caused by unexpected token.*

 - class [MissingTokenError](#)

Class for syntax error caused missing token.
 - class [SemanticError](#)

Base class for semantic errors.
 - class [ExpressionError](#)

Base class for errors of expression (e. g. type mismatch)
 - class [IndexError](#)

Error of index in array access expression.
 - class [RuleReferenceError](#)

Bad rule reference (referenced rule does not exists)
 - class [StringReferenceError](#)

Bad string reference (referenced string does not exists)
 - class [RangeError](#)

Bad range (e. g. left bound is greater than right bound)
 - class [FunctionCallError](#)

Bad function call (e. g. bad types or bad count of arguments)
 - class [SymbolReferenceError](#)

Bad symbol reference (e. g. reference to undefined symbol)
 - class [SymbolError](#)

Bad usage of symbol.
 - class [IncludeError](#)

Bad file include (e. g. included file does not exists)
 - class [StringError](#)

Error in string definition (e. g. collision of string identifiers)
 - class [StringModifierError](#)

Error of string modifier (e. g. invalid type of string modifier for specific string)
 - class [RegexError](#)

Regular expression error (e. g. bad repeat interval)
 - class [HexStrError](#)

Error in hexadecimal string (e. g. bad jump boundaries)
 - class [OverflowError](#)

Overflow error (e. g. integer overflow)
 - class [ModuleError](#)

Error that occurred in module import.
 - class [RuleError](#)

Error, that occurred in error definition (e. g. collision of rule identifiers)
 - class [RuleModifierError](#)

Error of rule modifier (e. g. duplicated rule modifier)
 - class [VariableError](#)

Error of variable definition.

6.5.1 Detailed Description

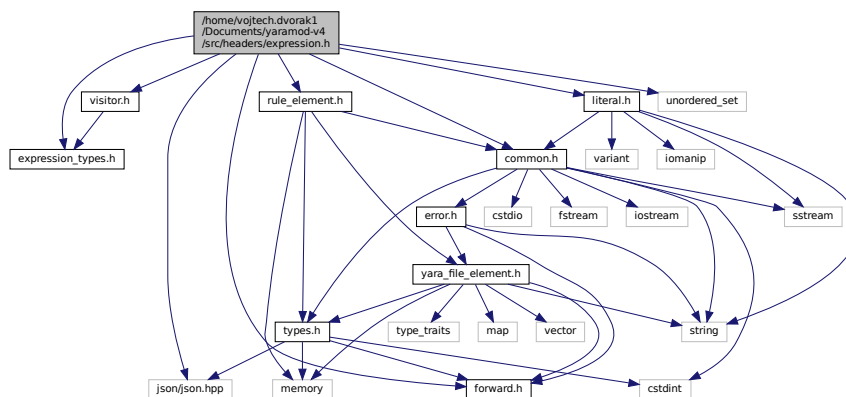
Contains declaration of error and exception hierarchies.

Author

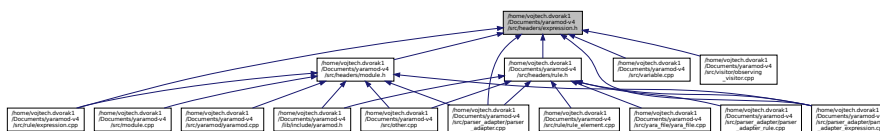
Vojtěch Dvořák

6.7 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expression.h File Reference

```
#include "visitor.h"
#include "common.h"
#include "json/json.hpp"
#include "literal.h"
#include "rule_element.h"
#include <unordered_set>
#include "forward.h"
#include "expression_types.h"
Include dependency graph for expression.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `Expression`
Represents expression in condition and in definition of internal variables.
- class `Range`
Class, that represents integer range (for example "(1..200)")
- class `Enum`
- class `ParenthesesExpression`
Base class for parentheses () in expressions.
- class `InExpression`
Class representing 'in' operator.
- class `BinaryExpression`
Class, that represents binary operators.
- class `MatchesExpression`

- class [AtExpression](#)
- class [BinaryArithmeticExpression](#)

Groups all binary arithmetics operators for more generic type checking.

- class [AddExpression](#)
- class [SubExpression](#)
- class [MulExpression](#)
- class [RemainderExpression](#)
- class [DivExpression](#)
- class [BinaryBitwiseExpression](#)

Groups all binary bitwise operators for more generic type checking.

- class [LeftShiftExpression](#)
- class [RightShiftExpression](#)
- class [BitwiseOrExpression](#)
- class [BitwiseAndExpression](#)
- class [BitwiseXorExpression](#)
- class [BinaryRelationalExpression](#)

Groups all binary bitwise operators for more generic type checking.

- class [LtExpression](#)
- class [LteExpression](#)
- class [GtExpression](#)
- class [GteExpression](#)
- class [EqExpression](#)
- class [NeqExpression](#)
- class [BinaryStrRelationalExpression](#)

Groups all binary bitwise operators for more generic type checking.

- class [ContainsExpression](#)
- class [IContainsExpression](#)
- class [StartsWithExpression](#)
- class [IStartsWithExpression](#)
- class [EndsWithExpression](#)
- class [IEndsWithExpression](#)
- class [IEqExpression](#)
- class [BinaryBoolExpression](#)

Groups all binary bitwise operators for more generic type checking.

- class [AndExpression](#)
- class [OrExpression](#)
- class [UnaryExpression](#)

Base class for unary operators (such as unary minus, negation...).

- class [DefinedExpression](#)
- class [UnaryArithmeticExpression](#)

Base class for all arithmetic operators (expressions), which have one operand.

- class [UnaryMinusExpression](#)
- class [UnaryBitwiseExpression](#)

Base class for bitwise unary operators (expressions), that have one operand. For example bitwise not (~) operator.

- class [BitwiseNotExpression](#)
- class [UnaryBoolExpression](#)

Base class for binary operators, that need boolean operands. For example and, or.

- class [NotExpression](#)
- class [TerminalExpression](#)

Base class for expression, that have not any subexpressions (operands)

- class [LiteralExpression](#)

Represents [Literal](#) in expression. Possible terminal node of expression tree (it does not have any other [Expression](#) operands).

- class [SizeExpression](#)
- class [RegexpExpression](#)

Represents regular expression literal in expressions.

- class [StringExpression](#)

Represents string identifier in expression ($\$<string_id>$)

- class [StringWildcardExpression](#)
- class [StringCountExpression](#)
- class [StringPropertyExpression](#)
- class [StringOffsetExpression](#)
- class [StringMatchLengthExpression](#)
- class [Symbol](#)

Base class for all symbols in expressions (structures, arrays, variables function calls...)

- class [ArrayExpression](#)
- class [StructExpression](#)
- class [FunctionCallExpression](#)
- class [PlainSymbol](#)
- class [VariableExpression](#)
- class [RuleWildcardExpression](#)
- class [QuantifierExpression](#)
- class [NoneExpression](#)

Represents 'none' keyword.

- class [AnyExpression](#)

Represents 'any' keyword.

- class [AllExpression](#)

Represents 'all' keyword.

- class [PercentQuantifierExpression](#)
- class [ThemExpression](#)

Represents 'them' keyword.

- class [SetExpression](#)

Base class for all set expressions.

- class [StringSetExpression](#)

Represents set of strings.

- class [RuleSetExpression](#)

Represents set of rules.

- class [ExpressionSetExpression](#)

Represents set of expressions.

- class [OfExpression](#)

Represents all variants of 'of' operator in YARA.

- class [ForExpression](#)

Represents 'for' operator that have rule set as iterable set.

- class [VarListExpression](#)

Represents list of variables, that are defined in 'for' expression for usage inside internal expression of 'for'.

- class [ForIntExpression](#)

Represents 'for' operator with defined variables for inner expression and with any iterable.

- struct [var_def_t](#)

Structure, that hold all information about in-expression defined variable.


```

graph TD
    Root["Normenobjekt.donors1  
Documentary object donors1"]
    Root --> Node1["Normenobjekt.donors1  
Documentary object donors1"]
    Root --> Node2["Normenobjekt.donors1  
Documentary object donors1"]
    Node1 --> Leaf1["Normenobjekt.donors1  
Documentary object donors1"]
    Node1 --> Leaf2["Normenobjekt.donors1  
Documentary object donors1"]
    Node1 --> Leaf3["Normenobjekt.donors1  
Documentary object donors1"]
    Node2 --> Leaf4["Normenobjekt.donors1  
Documentary object donors1"]
    Node2 --> Leaf5["Normenobjekt.donors1  
Documentary object donors1"]
    Node2 --> Leaf6["Normenobjekt.donors1  
Documentary object donors1"]
    Node2 --> Leaf7["Normenobjekt.donors1  
Documentary object donors1"]
    Node2 --> Leaf8["Normenobjekt.donors1  
Documentary object donors1"]
  
```

- class `FileContext`
Wrapper for all necessary data to perform incremental reparsing (they are concrete enough)
- struct `FileContext::cache_entry_t`
Entry in stored in cache.
- struct `FileContext::cache_t`
Cache of `FileContext` for offset `t` <-> `point_t` conversion.

Contains declaration of `FileContext` class.

Vojtěch Dvořák

Contains forward declarations of classes and structures.

Contains forward declarations of classes and structures.

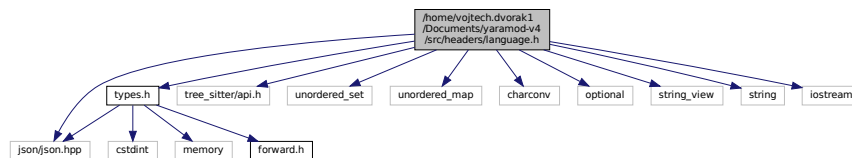
Vojtěch Dvořák

6.11 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/language.h File Reference

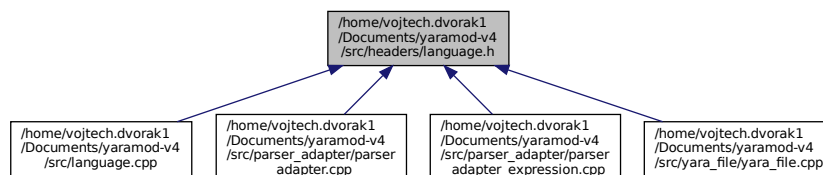
Declarations of functions and constants, that are associated with YARA language (table of keywords etc...)

```
#include "types.h"
#include "tree_sitter/api.h"
#include "json/json.hpp"
#include <unordered_set>
#include <unordered_map>
#include <charconv>
#include <optional>
#include <string_view>
#include <string>
#include <iostream>
```

Include dependency graph for language.h:



This graph shows which files directly or indirectly include this file:



Functions

- TSLanguage * [tree_sitter_yara](#) ()
- bool [isYaraKeyword](#) (std::string_view str)
- bool [isValidEscapeSequence](#) (std::string_view str)
- char [escapeSequenceToChar](#) (std::string_view str)
- const json * [getBuiltinSymbol](#) (std::string_view name)

6.11.1 Detailed Description

Declarations of functions and constants, that are associated with YARA language (table of keywords etc...)

Author

Vojtěch Dvořák

6.11.2 Function Documentation

6.11.2.1 escapeSequenceToChar()

```
char escapeSequenceToChar (
    std::string_view str )
```

Provides escape of escape sequences

Returns

character described by given escape sequence

< The hexa number in esc. seq. starts at index 2 and has length 2

< Convertibility should be ensured by the parser

6.11.2.2 getBuiltinSymbol()

```
const json* getBuiltinSymbol (
    std::string_view name )
```

Holds the static database of builtin symbols of the YARA language (int and uint functions, filesize variable)

Returns

JSON object with context of the symbol

6.11.2.3 isValidEscapeSequence()

```
bool isValidEscapeSequence (
    std::string_view str )
```

Checks if given string is valid escape sequence in YARA language

Returns

true if given string is valid escape sequence

< The check, if there is valid hexadecimal number after it is done by parser

6.11.2.4 isYaraKeyword()

```
bool isYaraKeyword (
    std::string_view str )
```

Determines whether given string_view is same as any keyword of YARA

Returns

true if given string is keyword of YARA language

6.11.2.5 tree_sitter_yara()

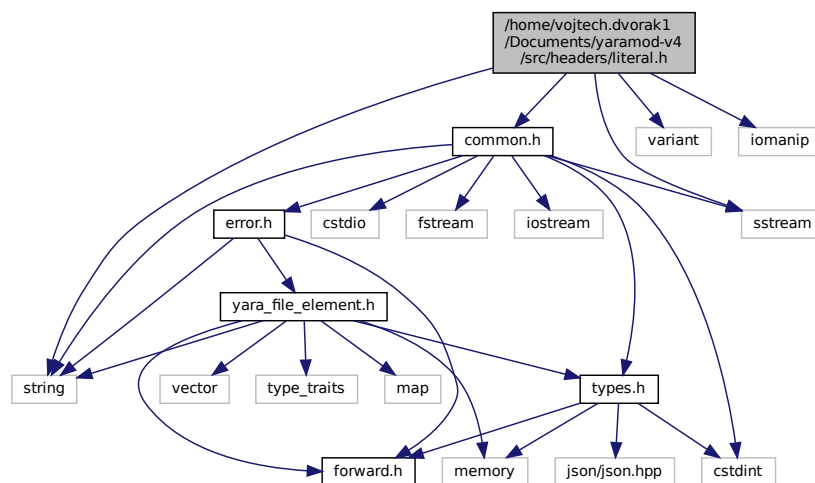
```
TSLanguage* tree_sitter_yara ( )
```

Function implemented by generated parser (in parser.c)

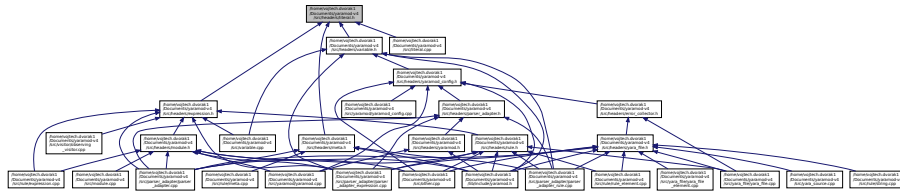
6.12 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/literal.h File Reference

Contains declaration of [Literal](#) class, that represents literals in [YaraFile](#).

```
#include "common.h"
#include <variant>
#include <sstream>
#include <string>
#include <iomanip>
Include dependency graph for literal.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Literal](#)

Object that represents literal, that may appear inside a rule definition.

Typedefs

- using **LiteralValue** = std::variant< std::monostate, std::string, int64_t, float, bool >

6.12.1 Detailed Description

Contains declaration of [Literal](#) class, that represents literals in [YaraFile](#).

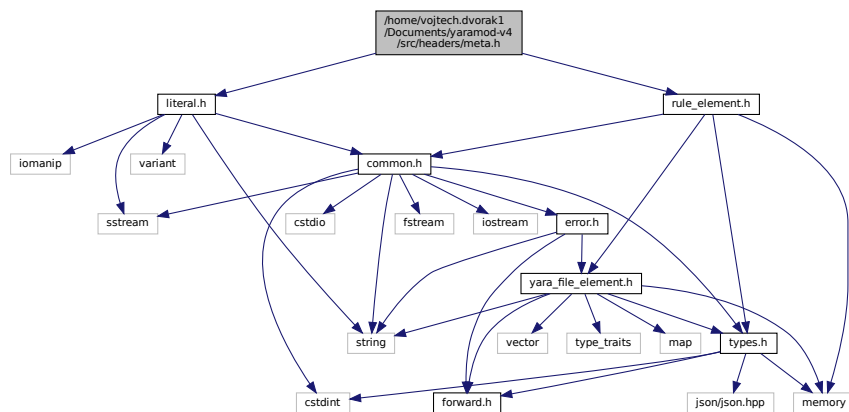
Author

Vojtěch Dvořák

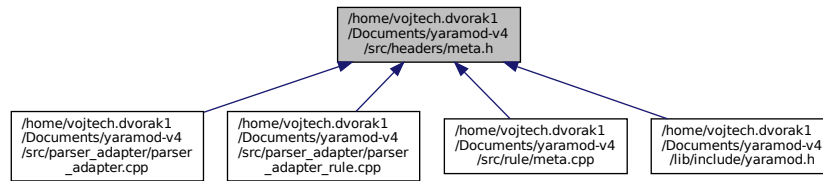
6.13 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/meta.h File Reference

Contains declaration of [Meta](#) class.

```
#include "literal.h"
#include "rule_element.h"
Include dependency graph for meta.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Meta](#)

Represents meta information of rule.

6.13.1 Detailed Description

Contains declaration of [Meta](#) class.

Author

Vojtěch Dvořák

6.14 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/module.h File Reference

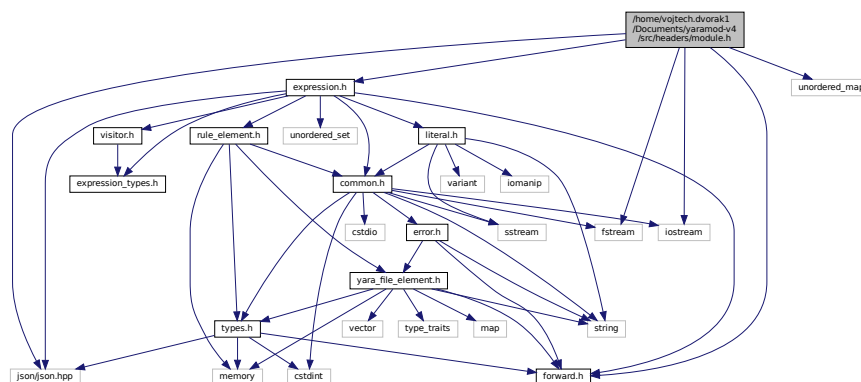
Classes for handling modules (their description in JSON) and their parsing.

```

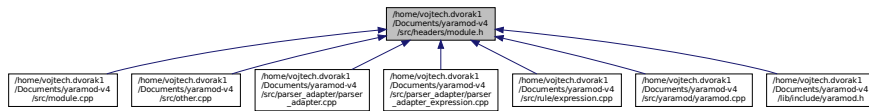
#include "json/json.hpp"
#include "expression.h"
#include <unordered_map>
#include <fstream>
#include <iostream>
#include "forward.h"

```

Include dependency graph for module.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ModuleProvider](#)
Singleton class that provides access to loaded modules. Modules are loaded, when [Yaramod](#) class is instantiated.
- class [Module](#)
Class representing module (its description)

Macros

- `#define` [USE_MODULE_CACHE](#)
[Comment](#) to disable using of module cache.

6.14.1 Detailed Description

Classes for handling modules (their description in JSON) and their parsing.

Author

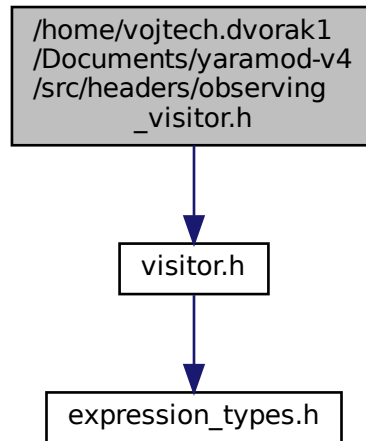
Vojtěch Dvořák

6.15 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/observing_visitor.h File Reference

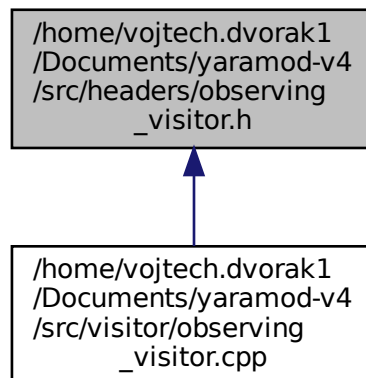
Contains declaration of [ObservingVisitor](#) class.

```
#include "visitor.h"
```

Include dependency graph for `observing_visitor.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [ObservingVisitor](#)

Concrete [Visitor](#) class for observing expression tree.

6.15.1 Detailed Description

Contains declaration of [ObservingVisitor](#) class.

Author

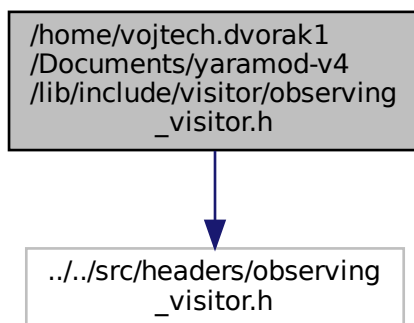
Vojtěch Dvořák

6.16 /home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/visitor/observing_visitor.h File Reference

Include file of expression observing visitor, interface, that can be use to inspection of YARA conditions or values of internal variables (Avast feature)

```
#include "../src/headers/observing_visitor.h"
```

Include dependency graph for observing_visitor.h:



6.16.1 Detailed Description

Include file of expression observing visitor, interface, that can be use to inspection of YARA conditions or values of internal variables (Avast feature)

Author

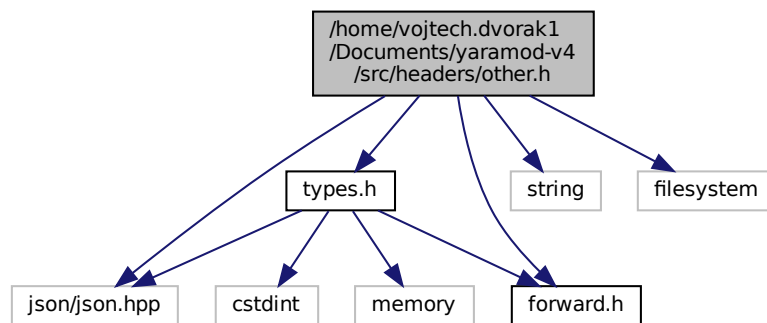
Vojtěch Dvořák

6.17 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/other.h File Reference

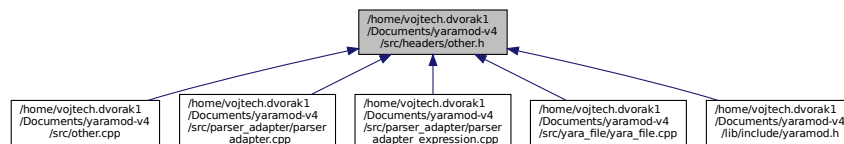
Classes, that represents less complex syntax structures (such as includes, imports, comments...)

```
#include "types.h"
#include <string>
#include <filesystem>
#include <json/json.hpp>
#include "forward.h"
```

Include dependency graph for other.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Comment](#)
Class representing comments (all valid types in YARA language)
- class [FileInclude](#)
Class representing "include <path>" structure in file.
- class [Import](#)
Class representing "import <module>" structure in file.

6.17.1 Detailed Description

Classes, that represents less complex syntax structures (such as includes, imports, comments...)

Author

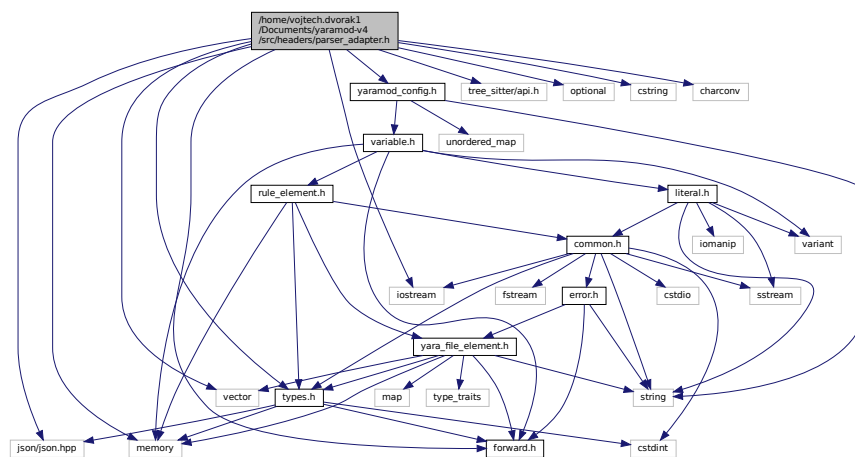
Vojtěch Dvořák

6.18 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/parser_adapter.h File Reference

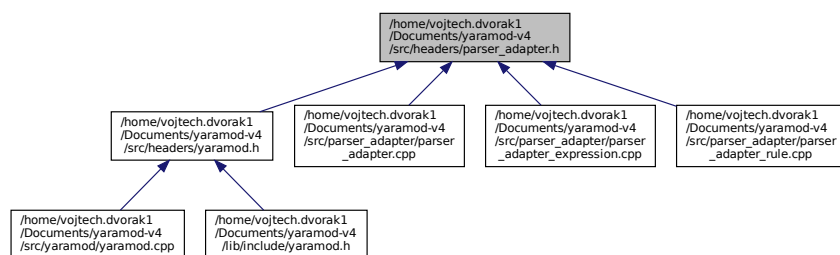
Header file containing abstract [IParserAdapter](#) class and concrete [TSParserAdapter](#) class.

```
#include "types.h"
#include "yaramod_config.h"
#include "json/json.hpp"
#include "tree_sitter/api.h"
#include <iostream>
#include <vector>
#include <optional>
#include <cstring>
#include <charconv>
#include <memory>
#include "forward.h"
```

Include dependency graph for parser_adapter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IParserAdapter](#)
Interface to which will be connected internal parser by concrete `ParserAdapter` (the "output socket" of the adapter)
- class [TSParserAdapter](#)
Class responsible for parsing of input string and converting TS structure to high level representation.

Macros

- `#define INITIAL_NODE_STACK_SIZE 128`
Initial size of auxiliary stack of nodes.

6.18.1 Detailed Description

Header file containing abstract `IParserAdapter` class and concrete `TSParserAdapter` class.

Author

Vojtěch Dvořák

6.18.2 Macro Definition Documentation

6.18.2.1 INITIAL_NODE_STACK_SIZE

```
#define INITIAL_NODE_STACK_SIZE 128
```

Initial size of auxiliary stack of nodes.

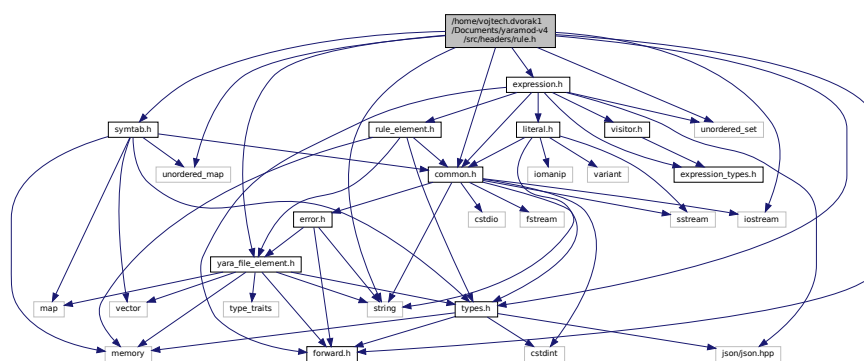
The capacity of stack is reserved when `TSParserAdapter` is instantiated, to save the time

6.19 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule.h File Reference

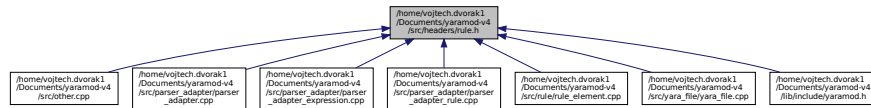
Contains declaration of class, that represents yara rule.

```
#include "types.h"
#include "yara_file_element.h"
#include "syntab.h"
#include "common.h"
#include "expression.h"
#include <string>
#include <unordered_map>
#include <unordered_set>
#include <iostream>
#include "forward.h"
```

Include dependency graph for rule.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [RuleModifier](#)
Class that represents rule modifiers.
- class [RuleModifierContainer](#)
- class [RuleModifierContainer::iterator](#)
Iterator for [RuleModifierContainer](#).
- class [Rule](#)
Class that represents yara rule.

6.19.1 Detailed Description

Contains declaration of class, that represents yara rule.

Contains class, that represents yara rule.

Author

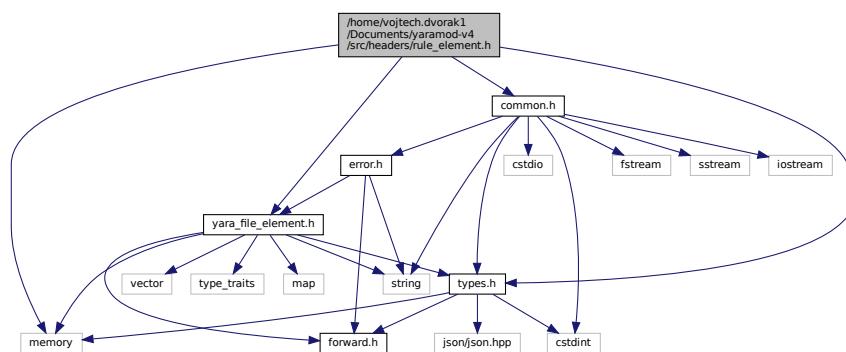
Vojtěch Dvořák

6.20 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule_element.h File Reference

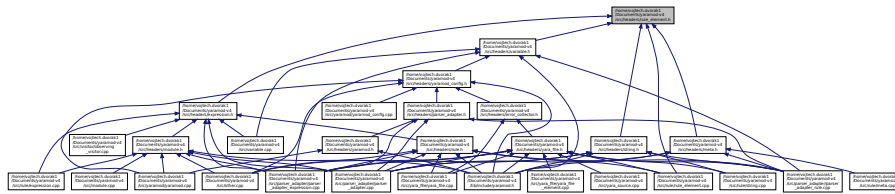
Contains declaration of [RuleElement](#) class.

```
#include "types.h"
#include "common.h"
#include "yara_file_element.h"
#include <memory>
```

Include dependency graph for rule_element.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [RuleElement](#)

Base class for all symbols that are valid in rule context.

6.20.1 Detailed Description

Contains declaration of [RuleElement](#) class.

Author

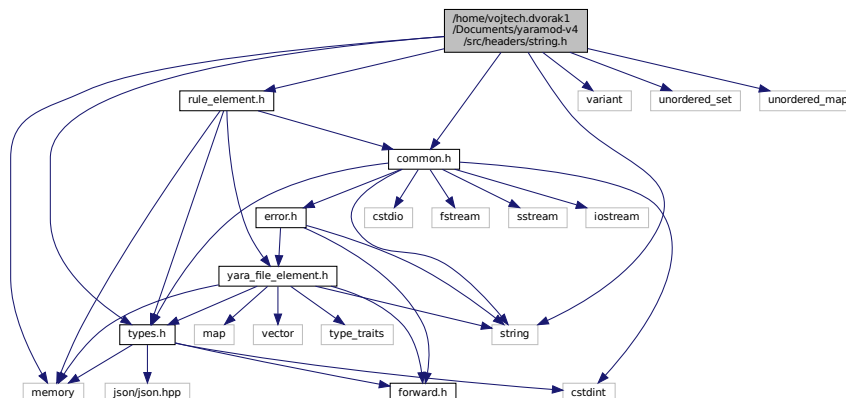
Vojtěch Dvořák

6.21 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/string.h File Reference

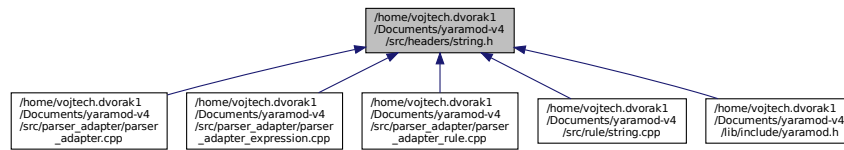
Declaration of classes [String](#), [StringModifier](#) and [StringModifierContainer](#).

```
#include "types.h"
#include "common.h"
#include "rule_element.h"
#include <string>
#include <memory>
#include <variant>
#include <unordered_set>
#include <unordered_map>
```

Include dependency graph for string.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [StringModifier](#)
Represents modifier of yara string.
- class [StringModifierContainer](#)
Iterable container for string modifier.
- class [StringModifierContainer::iterator](#)
Iterator for [StringModifierContainer](#).
- class [String](#)
Represents yara string.

Macros

- `#define` [BASE64_ALPHA_LEN](#) 64
- `#define` [XOR_UPPER_RANGE_BOUND](#) 255
- `#define` [REGEXP_MAXIMUM_QUANTIFIER_NUM](#) 32767

6.21.1 Detailed Description

Declaration of classes [String](#), [StringModifier](#) and [StringModifierContainer](#).

Author

Vojtěch Dvořák

6.21.2 Macro Definition Documentation

6.21.2.1 BASE64_ALPHA_LEN

```
#define BASE64_ALPHA_LEN 64
```

The length of base64 alphabet argument

Classes

- class [SymTab< V >](#)
Tab for symbols with double indexing.
- class [SymTab< V >::iterator](#)
SymTab iterator (just a wrapper above OffsetMap::const_iterator)
- class [SymTabDuplId< V >](#)
Almost same type of symbol table as the [SymTab](#), but in this case entries are allowed to have duplicated ids.
- class [SymTabDuplId< V >::Pool](#)
Pool that owns all elements with the same id.
- class [SymTabDuplId< V >::Pool::iterator](#)
Pool iterator.
- class [SymTabDuplId< V >::iterator](#)
SymTabDuplId iterator.
- class [SymTabOffset< V >](#)
Symbol tab for symbols that are indexed only by offsets (which can be duplicated)
- class [SymTabOffset< V >::iterator](#)
SymTabOffset iterator.

6.22.1 Detailed Description

Contains template classes for creating symbol tables.

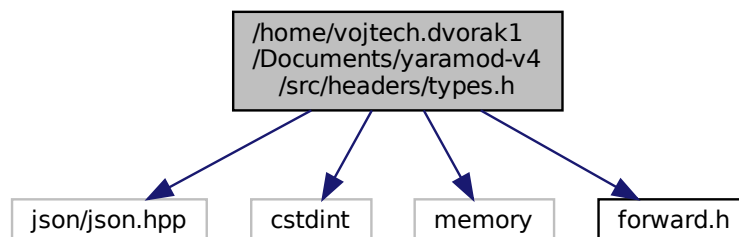
Author

Vojtěch Dvořák

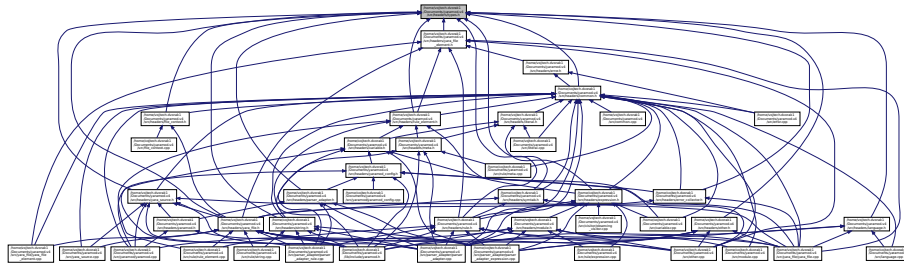
6.23 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/types.h File Reference

Declaration of structures and simple data types, that are used across the project.

```
#include "json/json.hpp"
#include <cstdint>
#include <memory>
#include "forward.h"
Include dependency graph for types.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [point_t](#)
Type for specifying position in input string.
- struct [range_t](#)
Range in input document specified by start point and end point.
- struct [string_hash_t](#)
Hash struct for [SymTab](#), provides heterogeneous comparison lookup.
- struct [offset_range_t](#)
Offset range for specifying the part of file, that was for example modified, deleted etc.
- struct [offset_edit_range_t](#)
Structure for lossless storing data about atomic edits in unified form.

Typedefs

- using **json** = `nlohmann::json`
- using **offset_t** = `uint32_t`
- using **RulePtr** = `std::unique_ptr< Rule >`
- using **ExpressionPtr** = `std::shared_ptr< Expression >`
- using **YaraFilePtr** = `std::shared_ptr< YaraFile >`
- using **YaraSourcePtr** = `std::unique_ptr< YaraSource >`

6.23.1 Detailed Description

Declaration of structures and simple data types, that are used across the project.

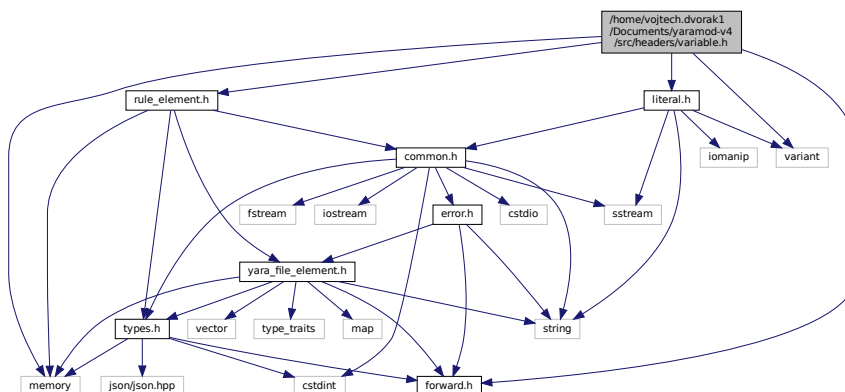
Author

Vojtěch Dvořák

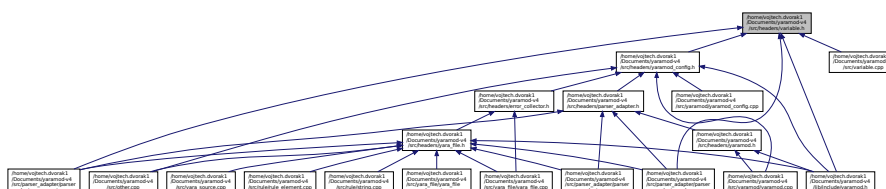
6.24 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/variable.h File Reference

Declaration of `Variable` class and its two specializations `IntVariable` (for internal variables, Avast feature of YARA) and `ExtVariables` (for external variables)

```
#include "literal.h"
#include "rule_element.h"
#include <variant>
#include <memory>
#include "forward.h"
Include dependency graph for variable.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class **Variable**
Base class for variables.
- class **IntVariable**
Class, that represents internal variables.
- class **ExtVariable**
Class that represents external variables.

6.24.1 Detailed Description

Declaration of [Variable](#) class and its two specializations [IntVariable](#) (for internal variables, Avast feature of YARA) and [ExtVariables](#) (for external variables)

Author

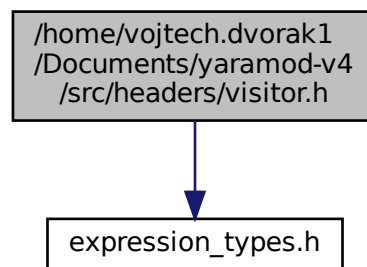
Vojtěch Dvořák

6.25 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/visitor.h File Reference

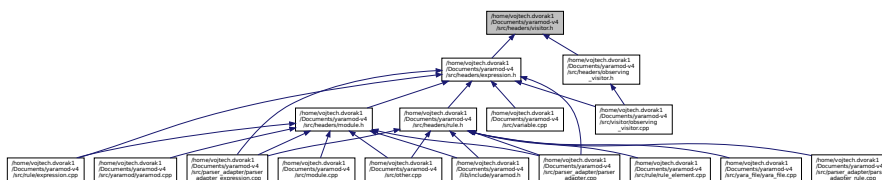
Contains declaration of [Visitor](#) class.

```
#include "expression_types.h"
```

Include dependency graph for visitor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Visitor](#)

Virtual base class for concrete visitor classes.

6.25.1 Detailed Description

Contains declaration of [Visitor](#) class.

Author

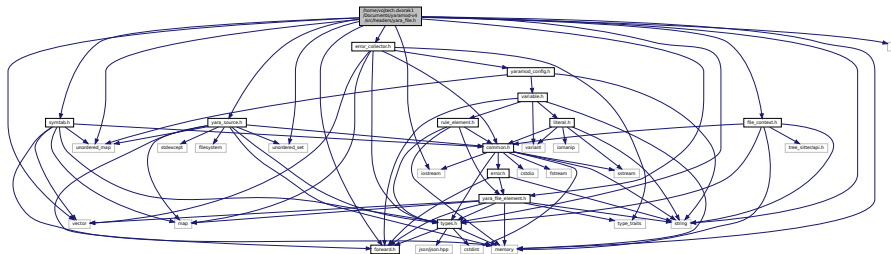
Vojtěch Dvořák

6.26 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_↵ file.h File Reference

Contains [YaraFile](#) class, that represents one continuous input in YARA language.

```
#include "yara_file_element.h"
#include "types.h"
#include "syntab.h"
#include "yara_source.h"
#include "error_collector.h"
#include "file_context.h"
#include <iostream>
#include <memory>
#include <string>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <vector>
#include "forward.h"
```

Include dependency graph for yara_file.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [edit_t](#)
Represents textual edit of [YaraFile](#).
- class [YaraFile](#)
Represents on continuous sequence of yara rules, file includes and imports.

6.26.1 Detailed Description

Contains [YaraFile](#) class, that represents one continuous input in YARA language.

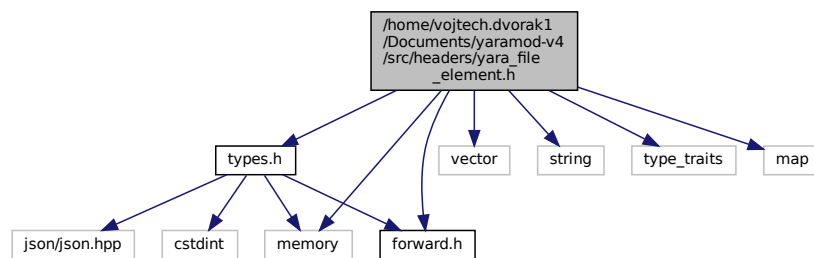
Author

Vojtěch Dvořák

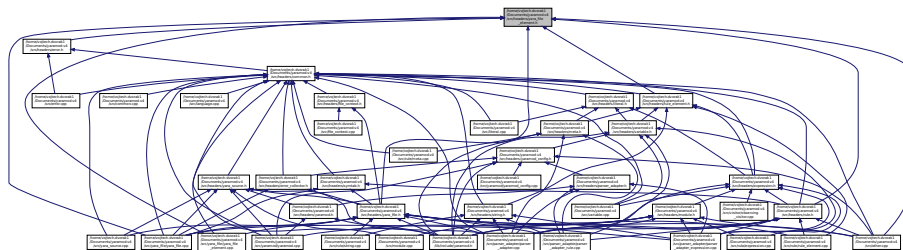
6.27 /home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file_element.h File Reference

Contains class, that represents syntax structures of [YaraFile](#).

```
#include "types.h"
#include <vector>
#include <string>
#include <type_traits>
#include <memory>
#include <map>
#include "forward.h"
Include dependency graph for yara_file_element.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [YaraFileElement](#)
Class representing syntax structures of [YaraFile](#) (such as Rules, Imports, Includes - everything which has some characters in the source string)
- class [YaraFileElementBindable](#)
Represents of elements of [YaraFile](#), that can have some semantic binding with other elements.
- class [TopLevelYaraFileElement](#)
Class for top level yara file elements.

Classes

- class [Yaramod](#)

The class, that controls parsing of Yara rulesets.

Macros

- `#define MODULE_PATH_ENV_VAR "YARAMODV4_MODULES"`

6.29.1 Detailed Description

Header file for class representing the parser of the library.

Author

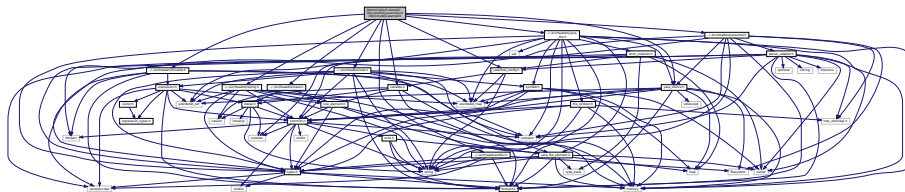
Vojtěch Dvořák

6.30 /home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/yaramod.h File Reference

Include file of yaramod-v4, that contains library interface for parsing of yara rulesets.

```
#include "../src/headers/yaramod.h"
#include "../src/headers/yaramod_config.h"
#include "../src/headers/yara_source.h"
#include "../src/headers/yara_file.h"
#include "../src/headers/variable.h"
#include "../src/headers/string.h"
#include "../src/headers/rule.h"
#include "../src/headers/meta.h"
#include "../src/headers/other.h"
#include "../src/headers/module.h"
```

Include dependency graph for yaramod.h:



6.30.1 Detailed Description

Include file of yaramod-v4, that contains library interface for parsing of yara rulesets.

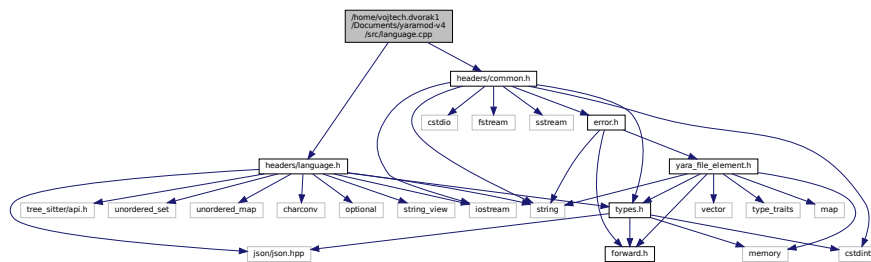
Author

Vojtěch Dvořák

6.32 /home/vojtech.dvorak1/Documents/yaramod-v4/src/language.cpp File Reference

Implementation of functions, that are associated with YARA.

```
#include "headers/common.h"
#include "headers/language.h"
Include dependency graph for language.cpp:
```



Functions

- bool [isYaraKeyword](#) (std::string_view str)
- bool [isValidEscapeSequence](#) (std::string_view str)
- char [escapeSequenceToChar](#) (std::string_view str)
- const json * [getBuiltinSymbol](#) (std::string_view name)

6.32.1 Detailed Description

Implementation of functions, that are associated with YARA.

Author

Vojtěch Dvořák

6.32.2 Function Documentation

6.32.2.1 escapeSequenceToChar()

```
char escapeSequenceToChar (
    std::string_view str )
```

Provides escape of escape sequences

Returns

character described by given escape sequence

< The hexa number in esc. seq. starts at index 2 and has length 2

< Convertibility should be ensured by the parser

6.32.2.2 getBuiltinSymbol()

```
const json* getBuiltinSymbol (
    std::string_view name )
```

Holds the static database of builtin symbols of the YARA language (int and uint functions, filesize variable)

Returns

JSON object with context of the symbol

6.32.2.3 isValidEscapeSequence()

```
bool isValidEscapeSequence (
    std::string_view str )
```

Checks if given string is valid escape sequence in YARA language

Returns

true if given string is valid escape sequence

< The check, if there is valid hexadecimal number after it is done by parser

6.32.2.4 isYaraKeyword()

```
bool isYaraKeyword (
    std::string_view str )
```

Determines whether given string_view is same as any keyword of YARA

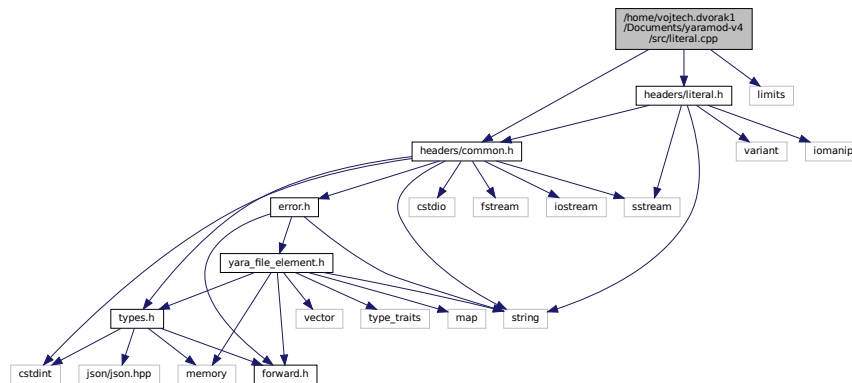
Returns

true if given string is keyword of YARA language

6.33 /home/vojtech.dvorak1/Documents/yaramod-v4/src/literal.cpp File Reference

Implementation class [Literal](#) methods.

```
#include "headers/common.h"
#include "headers/literal.h"
#include <limits>
Include dependency graph for literal.cpp:
```



6.33.1 Detailed Description

Implementation class [Literal](#) methods.

Author

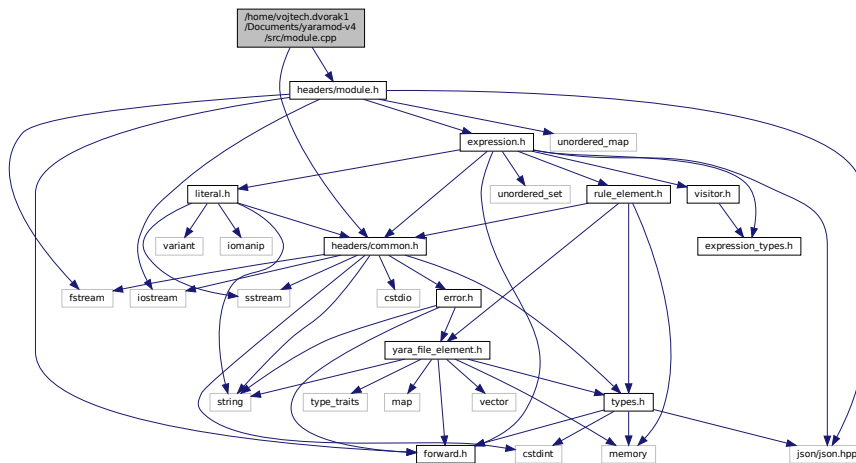
Vojtěch Dvořák

6.34 /home/vojtech.dvorak1/Documents/yaramod-v4/src/module.cpp File Reference

Implementation of members of [Module](#) and [ModuleProvider](#).

```
#include "headers/common.h"
#include "headers/module.h"
```

Include dependency graph for module.cpp:



6.34.1 Detailed Description

Implementation of members of [Module](#) and [ModuleProvider](#).

Author

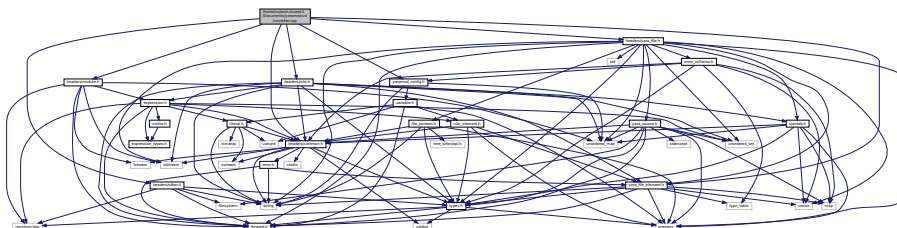
Vojtěch Dvořák

6.35 /home/vojtech.dvorak1/Documents/yaramod-v4/src/other.cpp File Reference

Implementation of methods that belongs to "other" elements of [YaraFile](#) (includes, imports and comments)

```
#include "headers/common.h"
#include "headers/module.h"
#include "headers/yara_file.h"
#include "headers/rule.h"
#include "headers/yara_file_element.h"
#include "headers/yaramod_config.h"
#include "headers/other.h"
```

Include dependency graph for other.cpp:



6.35.1 Detailed Description

Implementation of methods that belongs to "other" elements of [YaraFile](#) (includes, imports and comments)

Author

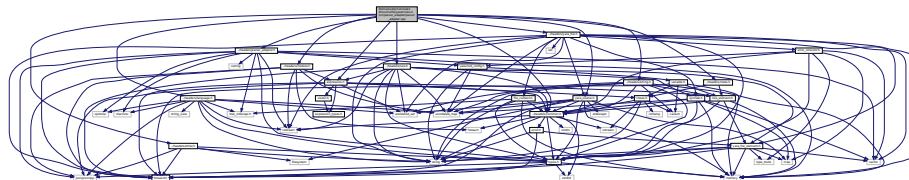
Vojtěch Dvořák

6.36 /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter.cpp File Reference

Implementation of [TSParserAdapter](#) class members.

```
#include "../headers/language.h"
#include "../headers/common.h"
#include "../headers/meta.h"
#include "../headers/rule.h"
#include "../headers/yara_file.h"
#include "../headers/yara_source.h"
#include "../headers/other.h"
#include "../headers/module.h"
#include "../headers/string.h"
#include "../headers/expression.h"
#include "../headers/parser_adapter.h"
```

Include dependency graph for parser_adapter.cpp:



Classes

- struct [node_w_depth_t](#)

6.36.1 Detailed Description

Implementation of [TSParserAdapter](#) class members.

Class [TSParserAdapter](#) is responsible for converting concrete syntax tree from tree sitter to high level AST

Note: Originally it was planned to use tree-sitter builtin mechanism for analysis of concrete syntax tree - TS queries, but it looks like, that they are not so effective like more adhoc and less universal methods (simple traversal of tree)

Author

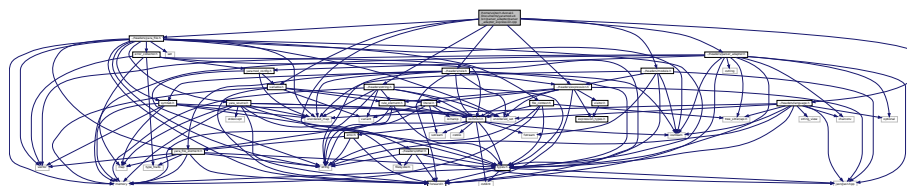
Vojtěch Dvořák

6.37 /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter_expression.cpp File Reference

Definitions of methods, that are responsible for converting concrete syntax tree returned by tree-sitter to high level representation.

```
#include "../headers/yara_file.h"
#include "../headers/expression.h"
#include "../headers/rule.h"
#include "../headers/other.h"
#include "../headers/string.h"
#include "../headers/language.h"
#include "../headers/variable.h"
#include "../headers/module.h"
#include "../headers/parser_adapter.h"
```

Include dependency graph for parser_adapter_expression.cpp:



6.37.1 Detailed Description

Definitions of methods, that are responsible for converting concrete syntax tree returned by tree-sitter to high level representation.

Author

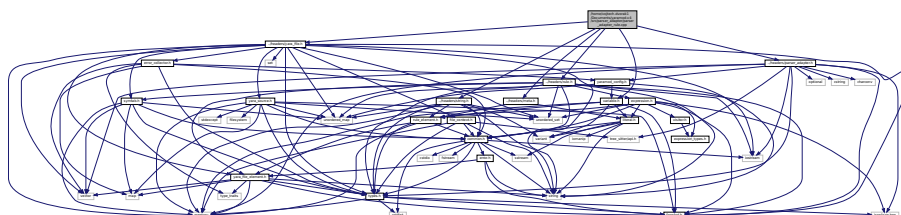
Vojtěch Dvořák

6.38 /home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter_rule.cpp File Reference

Implementation of [TSParserAdapter](#) members, that are responsible for converting TSNode with rules and rule elements to high level representation

```
#include "../headers/yara_file.h"
#include "../headers/rule.h"
#include "../headers/string.h"
#include "../headers/meta.h"
#include "../headers/variable.h"
#include "../headers/parser_adapter.h"
```

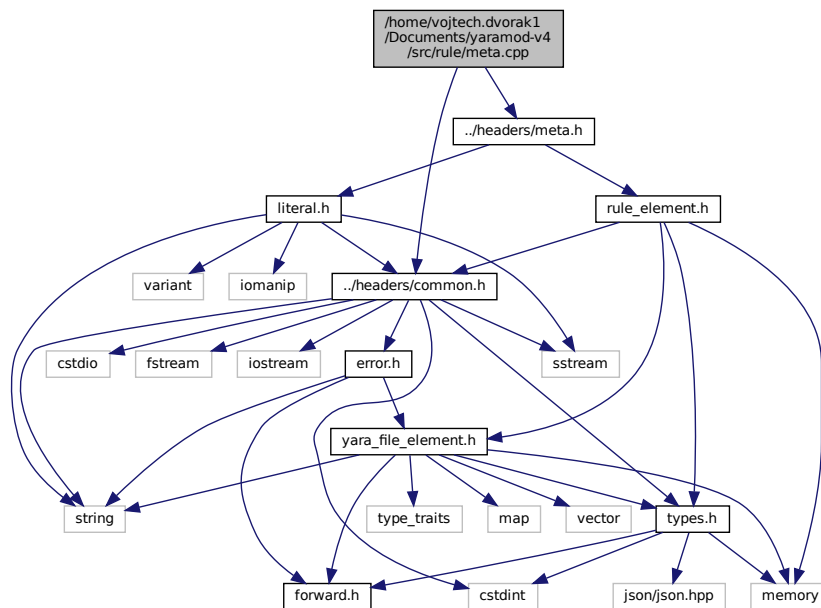
Include dependency graph for parser_adapter_rule.cpp:



6.40 /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/meta.cpp File Reference

Contains implementation of methods of [Meta](#).

```
#include "../headers/common.h"
#include "../headers/meta.h"
Include dependency graph for meta.cpp:
```



6.40.1 Detailed Description

Contains implementation of methods of [Meta](#).

Author

Vojtěch Dvořák

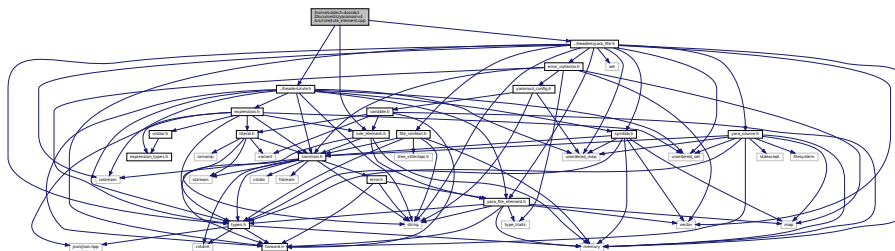
6.41 /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/rule_↵ element.cpp File Reference

Implementation of [RuleElement](#) methods.

```
#include "../headers/rule.h"
#include "../headers/yara_file.h"
```



```
#include "../headers/rule_element.h"
Include dependency graph for rule_element.cpp:
```



6.41.1 Detailed Description

Implementation of **RuleElement** methods.

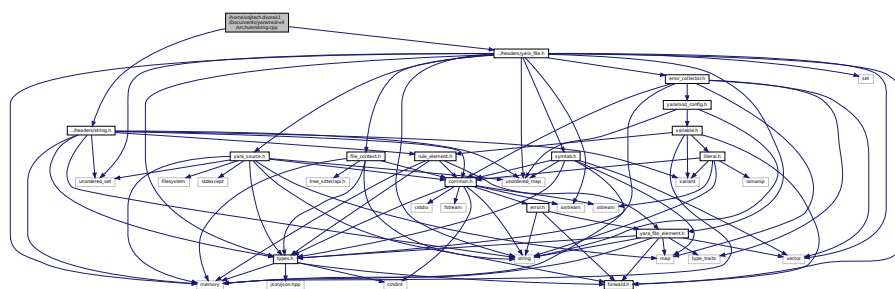
Author

Vojtěch Dvořák

6.42 /home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/string.cpp

Implementation of String and StringModifier methods.

```
#include "../headers/string.h"
#include "../headers/yara_file.h"
Include dependency graph for string.cpp:
```



6.42.1 Detailed Description

Implementation of String and StringModifier methods.

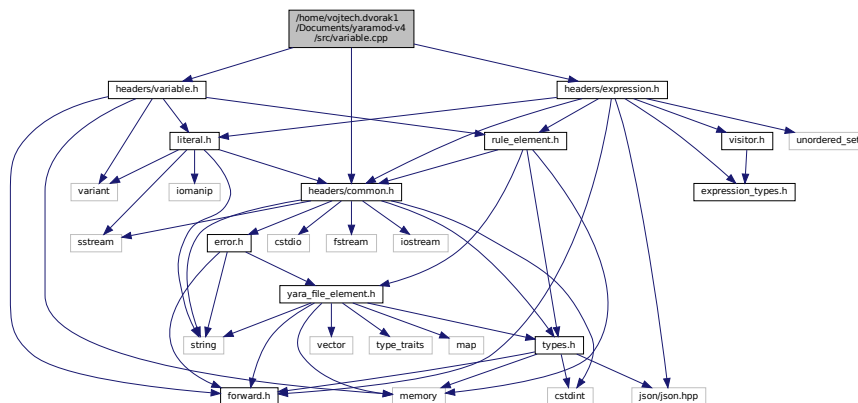
Author

Vojtěch Dvořák

6.43 /home/vojtech.dvorak1/Documents/yaramod-v4/src/variable.cpp File Reference

Implementation of [Variable](#), [IntVariable](#), [ExtVariable](#) methods.

```
#include "headers/common.h"
#include "headers/expression.h"
#include "headers/variable.h"
Include dependency graph for variable.cpp:
```



6.43.1 Detailed Description

Implementation of [Variable](#), [IntVariable](#), [ExtVariable](#) methods.

Author

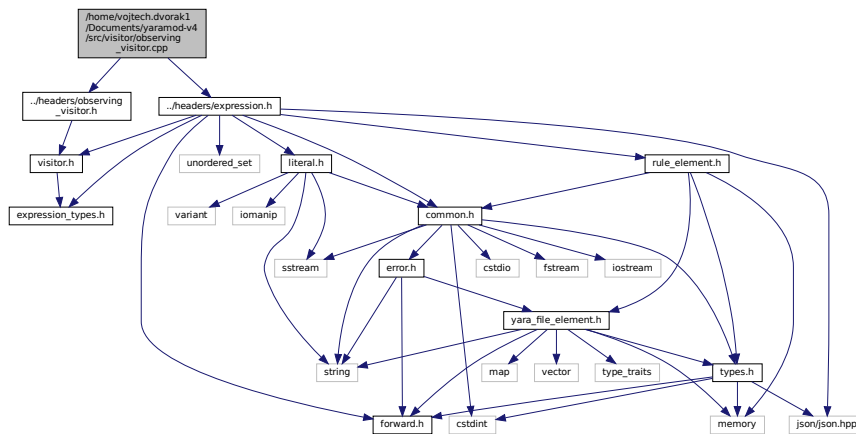
Vojtěch Dvořák

6.44 /home/vojtech.dvorak1/Documents/yaramod-v4/src/visitor/observing_visitor.cpp File Reference

Contains default implementation of [ObservingVisitor](#) methods.

```
#include "../headers/expression.h"
#include "../headers/observing_visitor.h"
```

Include dependency graph for `observing_visitor.cpp`:



6.44.1 Detailed Description

Contains default implementation of [ObservingVisitor](#) methods.

Author

Vojtěch Dvořák

6.45 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file.cpp File Reference

Implementation of [YaraFile](#) methods.

```

#include "../headers/common.h"
#include "../headers/yara_file_element.h"
#include "../headers/rule.h"
#include "../headers/language.h"
#include "../headers/error_collector.h"
#include "../headers/other.h"
#include "../headers/symtab.h"
#include "../headers/yara_source.h"
#include "../headers/yara_file.h"

```

Include dependency graph for `yara_file.cpp`:



Functions

- `template<typename T >`
`bool hitsRange (const T &element, const offset_range_t &range, offset_range_t &max_range)`
- `template<typename T >`
`void shift (const T &element, const int32_t &delta)`
- `template<typename T >`
`void shiftCachedRange (const T &element, const uint32_t &start_row, const int32_t &row_delta, const int32_t &col)`
- `void updateModifiedRanges (const edit_t &e, std::vector< offset_edit_range_t > &ranges)`
- `void computePositionShift (const edit_t &e, int32_t &row_diff, int32_t &col)`

6.45.1 Detailed Description

Implementation of [YaraFile](#) methods.

Author

Vojtěch Dvořák

6.45.2 Function Documentation

6.45.2.1 updateModifiedRanges()

```
void updateModifiedRanges (
    const edit\_t & e,
    std::vector< offset\_edit\_range\_t > & ranges )
```

< If previous edit was deletion, it is processed differently, because there are no characters in deleted range

< New edit occurs completely after older edit (older edit is not affect by the new edit)

< New edit occurs inside older edit range -> extend or shrink edited interval

< New edit occurs completely after older edit (older edit is not affect by the new edit)

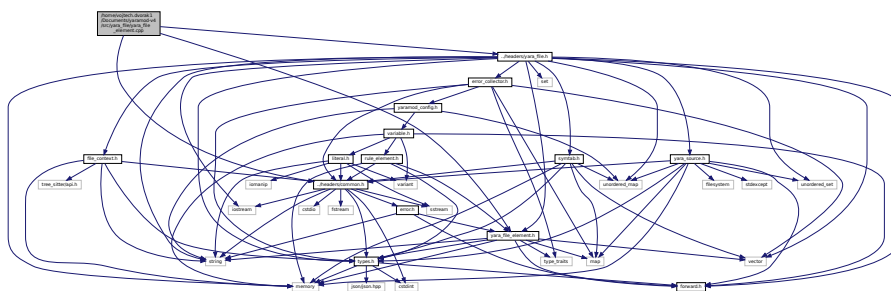
< New edit start occurs before older edit

< New edit completely occurs before older edit

6.46 `/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file_element.cpp` File Reference

Implementation of YaraFileElement methods.

```
#include "../headers/common.h"
#include "../headers/yara_file.h"
#include "../headers/yara_file_element.h"
Include dependency graph for yara_file_element.cpp:
```



6.46.1 Detailed Description

Implementation of YaraFileElement methods.

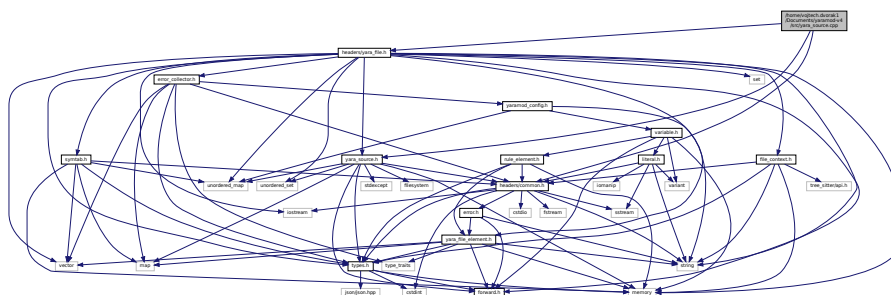
Author

Vojtěch Dvořák

6.47 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_↔ source.cpp File Reference

Implementation of YaraSource methods.

```
#include "headers/common.h"
#include "headers/yara_file.h"
#include "headers/yara_source.h"
Include dependency graph for yara_source.cpp:
```

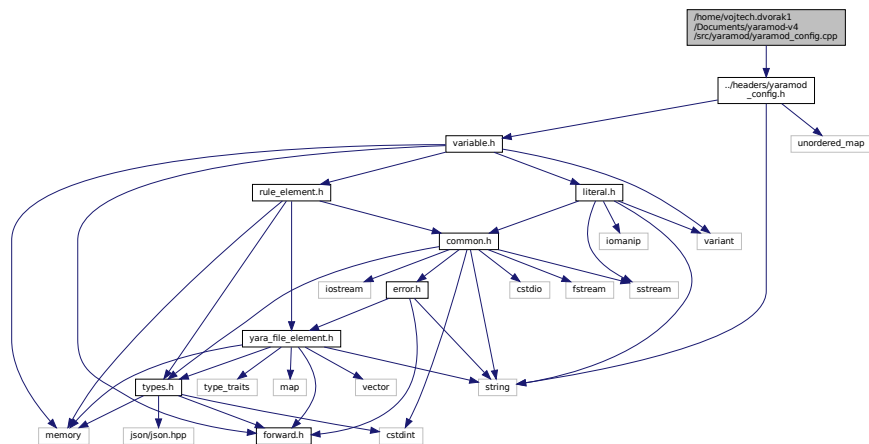


6.49 /home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_config.cpp File Reference

Contains implementation of methods of YaraModConfig.

```
#include "../headers/yaramod_config.h"
```

Include dependency graph for yaramod_config.cpp:



6.49.1 Detailed Description

Contains implementation of methods of YaraModConfig.

Author

Vojtěch Dvořák

Index

/home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/visitor/observing_visitor.h, 457
/home/vojtech.dvorak1/Documents/yaramod-v4/lib/include/yaramod.h, 473
/home/vojtech.dvorak1/Documents/yaramod-v4/src/common.cpp, 435
/home/vojtech.dvorak1/Documents/yaramod-v4/src/error.cpp, 437
/home/vojtech.dvorak1/Documents/yaramod-v4/src/file_context.cpp, 437
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/common.h, 438
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error.h, 441
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/error_context.h, 444
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expressions.h, 445
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/expressions_types.h, 448
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/file_context.h, 448
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/forward.h, 449
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/language.h, 450
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/literal.h, 452
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/meta.h, 453
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/module.h, 454
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/observing_visitor.h, 455
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/other.h, 458
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/parser_adapter.h, 459
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule.h, 460
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/rule_element.h, 461
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/string.h, 462
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/symtab.h, 464
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/types.h, 465
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/variable.h, 469
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/visitor/observing_visitor.h, 457
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file/yara_file.h, 468
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_file_element.h, 469
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yara_source.h, 470
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yaramod.h, 472
/home/vojtech.dvorak1/Documents/yaramod-v4/src/headers/yaramod_context.h, 473
/home/vojtech.dvorak1/Documents/yaramod-v4/src/language.cpp, 475
/home/vojtech.dvorak1/Documents/yaramod-v4/src/literal.cpp, 476
/home/vojtech.dvorak1/Documents/yaramod-v4/src/module.cpp, 477
/home/vojtech.dvorak1/Documents/yaramod-v4/src/other.cpp, 478
/home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter.cpp, 479
/home/vojtech.dvorak1/Documents/yaramod-v4/src/parser_adapter/parser_adapter.h, 480
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/expression.cpp, 481
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/meta.cpp, 482
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/rule_element.cpp, 483
/home/vojtech.dvorak1/Documents/yaramod-v4/src/rule/string.cpp, 484
/home/vojtech.dvorak1/Documents/yaramod-v4/src/variable.cpp, 485
/home/vojtech.dvorak1/Documents/yaramod-v4/src/visitor/observing_visitor.cpp, 486
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file.cpp, 487
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_file/yara_file_element.cpp, 488
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yara_source.cpp, 489
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod.cpp, 490
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_context.cpp, 491
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_header.cpp, 492
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_parser.cpp, 493
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_rule.cpp, 494
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_variable.cpp, 495
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_visitor.cpp, 496
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_context.h, 497
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_header.h, 498
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_parser.h, 499
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_rule.h, 500
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_variable.h, 501
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_visitor.h, 502
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_context.cpp, 497
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_header.cpp, 498
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_parser.cpp, 499
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_rule.cpp, 500
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_variable.cpp, 501
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_visitor.cpp, 502
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_context.h, 497
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_header.h, 498
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_parser.h, 499
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_rule.h, 500
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_variable.h, 501
/home/vojtech.dvorak1/Documents/yaramod-v4/src/yaramod/yaramod_visitor.h, 502

- accept
 - AddExpression, 18
 - AllExpression, 20
 - AndExpression, 22
 - AnyExpression, 24
 - ArrayExpression, 26
 - AtExpression, 29
 - BitwiseAndExpression, 46
 - BitwiseNotExpression, 48
 - BitwiseOrExpression, 50
 - BitwiseXorExpression, 52
 - ContainsExpression, 58
 - DefinedExpression, 60
 - DivExpression, 63
 - EndsWithExpression, 66
 - Enum, 68
 - EqExpression, 71
 - Expression, 80
 - ExpressionSetExpression, 86
 - ForExpression, 95
 - ForIntExpression, 99
 - FunctionCallExpression, 106
 - GteExpression, 110
 - GtExpression, 112
 - IContainsExpression, 118
 - IEndsWithExpression, 120
 - IEqExpression, 122
 - InExpression, 134
 - IStartsWithExpression, 143
 - LeftShiftExpression, 150
 - LiteralExpression, 155
 - LteExpression, 158
 - LtExpression, 160
 - MatchesExpression, 162
 - MulExpression, 181
 - NeqExpression, 183
 - NoneExpression, 186
 - NotExpression, 188
 - OfExpression, 193
 - OrExpression, 198
 - ParenthesesExpression, 204
 - PercentQuantifierExpression, 207
 - PlainSymbol, 210
 - Range, 219
 - RegexpExpression, 231
 - RemainderExpression, 234
 - RightShiftExpression, 237
 - RuleSetExpression, 266
 - RuleWildcardExpression, 269
 - SizeExpression, 277
 - StartsWithExpression, 280
 - StringCountExpression, 287
 - StringExpression, 293
 - StringMatchLengthExpression, 296
 - StringOffsetExpression, 308
 - StringSetExpression, 317
 - StringWildcardExpression, 320
 - StructExpression, 322
 - SubExpression, 325
 - ThemExpression, 354
 - UnaryMinusExpression, 374
 - VariableExpression, 386
 - VarListExpression, 388
- add
 - ErrorCollector< E >, 74
 - RuleModifierContainer, 254
 - StringModifierContainer, 301
- addBoolVar
 - YaramodConfig, 422
- addComment
 - YaraFile, 396
- addEdit
 - YaraFile, 396
- addError
 - YaraFile, 396
- AddExpression, 17
 - accept, 18
 - opsign, 18
- addFile
 - YaraSource, 428
- addFileInclude
 - YaraFile, 396
- addImport
 - YaraFile, 397
- addMeta
 - Rule, 240
- addMissingSymbol
 - YaraFile, 397
- addModifier
 - Rule, 240
 - String, 283
- addModule
 - ModuleProvider, 178
- addRule
 - YaraFile, 397
- addSemanticError
 - YaraFile, 397
- addString
 - Rule, 240
- addSyntaxError
 - YaraFile, 397
- addTag
 - Rule, 240
- addVar
 - Rule, 241
 - YaramodConfig, 423
- addWrong
 - YaraFile, 398
- AllExpression, 19
 - accept, 20
 - getTextFormatted, 20
- AndExpression, 21
 - accept, 22
 - opsign, 22
- AnyExpression, 23
 - accept, 24

- getFormatted, 24
- areOperandsValid
 - ArrayExpression, 26
 - BinaryExpression, 39
 - Enum, 68
 - Expression, 80
 - ForExpression, 95
 - ForIntExpression, 99
 - FunctionCallExpression, 107
 - InExpression, 134
 - OfExpression, 193
 - ParenthesesExpression, 204
 - PercentQuantifierExpression, 207
 - QuantifierExpression, 216
 - Range, 219
 - SetExpression, 274
 - StringPropertyExpression, 310
 - StructExpression, 323
 - Symbol, 328
 - TerminalExpression, 352
 - ThemExpression, 354
 - UnaryExpression, 371
 - VarListExpression, 388
- ArrayExpression, 25
 - accept, 26
 - areOperandsValid, 26
 - getFormatted, 27
 - isComplete, 27
 - isValid, 27
- at
 - SymTabDuplId < V >, 344
- AtExpression, 28
 - accept, 29
 - getType, 29
 - isValid, 30
 - opsign, 30
- Auto
 - YaramodConfig, 422
- BASE64_ALPHA_LEN
 - string.h, 463
- begin
 - RuleModifierContainer, 254
 - StringModifierContainer, 301
- BinaryArithmeticExpression, 31
 - getType, 32
 - isValid, 32
- BinaryBitwiseExpression, 33
 - getType, 34
 - isValid, 34
- BinaryBoolExpression, 35
 - getType, 36
 - isValid, 36
- BinaryExpression, 37
 - areOperandsValid, 39
 - getFormatted, 40
 - isComplete, 40
 - opsign, 40
- BinaryRelationalExpression, 41
 - getType, 42
 - isValid, 42
- BinaryStrRelationalExpression, 43
 - getType, 44
 - isValid, 44
- bind
 - YaraFile, 398
 - YaraFileElementBindable, 417
 - YaraSource, 428
- bind_to
 - YaraFileElementBindable, 417
- BitwiseAndExpression, 45
 - accept, 46
 - opsign, 46
- BitwiseNotExpression, 47
 - accept, 48
- BitwiseOrExpression, 49
 - accept, 50
 - opsign, 50
- BitwiseXorExpression, 51
 - accept, 52
 - opsign, 52
- buildModuleContext
 - Module, 170
- cacheSymbol
 - ModuleProvider, 178
- checkDuplicatedRules
 - YaraFile, 398
- checkMissing
 - YaraFile, 398
- checkRange
 - YaraFile, 399
- clear
 - ErrorCollector < E >, 74
 - SymTab < V >, 340
 - SymTabDuplId < V >, 344
- clearAll
 - YaraFile, 399
- clearCache
 - ModuleProvider, 178
- clearComments
 - YaraFile, 399
- clearEdits
 - YaraFile, 399
- clearInactiveFiles
 - YaraSource, 428
- clearLocalFileIncludes
 - YaraFile, 399
- clearLocalImports
 - YaraFile, 399
- clearLocalRules
 - YaraFile, 399
- clearSemanticErrors
 - YaraFile, 400
- clearSyntaxErrors
 - YaraFile, 400
- clearToBeChecked
 - YaraFile, 400

- clearTree
 - FileContext, 89
- clearWrong
 - YaraFile, 400
- Comment, 55
 - getTextFormatted, 56
- common.cpp
 - normalizeFSPath, 436
 - offsetToPoint, 436
 - pointToOffset, 436
- common.h
 - normalizeFSPath, 440
 - offsetToPoint, 440
 - pointToOffset, 440
- config
 - Yaramod, 418
- ContainsExpression, 57
 - accept, 58
 - opsign, 58
- createRuleError
 - YaraFile, 400
- createYaraSource
 - IParserAdapter, 140
 - TSParserAdapter, 359
- createYaraSourceFromFile
 - IParserAdapter, 140
 - TSParserAdapter, 359
- ctx
 - YaraFile, 400
- ctx_
 - Symbol, 330
- dataById
 - SymTab< V >, 341
- dataByOffset
 - SymTab< V >, 341
- deactivateFiles
 - YaraSource, 428
- defined_
 - Symbol, 330
- DefinedExpression, 59
 - accept, 60
 - getType, 60
 - isValid, 61
- DEL
 - offset_edit_range_t, 196
- deleteBindings
 - YaraFile, 400
 - YaraSource, 428
- diffTree
 - FileContext, 89
- DivExpression, 62
 - accept, 63
 - opsign, 63
- dtype_
 - Symbol, 331
- dump
 - ErrorCollector< E >, 74
- dumpBindChildren
 - YaraSource, 428
- dumpBindParents
 - YaraSource, 429
- dumpSyntaxErrors
 - YaraFile, 401
- edit
 - YaraFile, 401
 - YaraSource, 429, 430
- edit_t, 64
- EditType
 - offset_edit_range_t, 195
- empty
 - RuleModifierContainer, 254
 - StringModifierContainer, 301
- end
 - RuleModifierContainer, 255
 - StringModifierContainer, 301
- EndsWithExpression, 65
 - accept, 66
 - opsign, 66
- Enum, 67
 - accept, 68
 - areOperandsValid, 68
 - getTextFormatted, 69
 - getType, 69
 - isComplete, 69
 - isValid, 69
- EqExpression, 70
 - accept, 71
 - opsign, 71
- erase
 - RuleModifierContainer, 255
 - StringModifierContainer, 301
 - SymTabDuplId< V >::Pool, 212
- Error, 72
 - exception, 73
 - getDescription, 73
- ErrorCollector< E >, 73
 - add, 74
 - clear, 74
 - dump, 74
 - get, 75
 - getConst, 75
 - remove, 75
 - revRemoveUntil, 75
 - update, 75
- ErrorException, 76
- ErrorMode
 - YaramodConfig, 421
- escapeSequenceToChar
 - language.cpp, 475
 - language.h, 451
- exception
 - Error, 73
- exists
 - SymTabDuplId< V >, 344
 - SymTabOffset< V >, 347
- Expression, 77

- accept, [80](#)
- areOperandsValid, [80](#)
- getType, [80](#)
- isArithmeticCompatible, [81](#)
- isComplete, [81](#)
- isValid, [81](#)
- typeToString, [82](#)
- ExpressionError, [82](#)
- ExpressionErrorException, [83](#)
- ExpressionSetExpression, [85](#)
 - accept, [86](#)
 - getTextFormatted, [87](#)
- ExtVariable, [87](#)
 - getValue, [88](#)
 - setValue, [88](#)
- FileContext, [89](#)
 - clearTree, [89](#)
 - diffTree, [89](#)
 - getString, [90](#)
 - getTempString, [90](#)
 - getTree, [90](#)
 - setString, [90](#)
 - setTree, [90](#)
 - stringEdit, [90](#)
 - stringUpdate, [91](#)
 - toTSEdit, [91](#)
 - treeEdit, [91](#)
 - undoEdits, [91](#)
- FileContext::cache_entry_t, [53](#)
- FileContext::cache_t, [54](#)
- FileInclude, [92](#)
 - getFile, [93](#)
 - getId, [93](#)
 - getPath, [93](#)
 - getTextFormatted, [93](#)
- findCached
 - ModuleProvider, [179](#)
- findField
 - Module, [170](#)
- findInArray
 - Module, [170](#)
- ForExpression, [94](#)
 - accept, [95](#)
 - areOperandsValid, [95](#)
 - getTextFormatted, [95](#)
 - getType, [96](#)
 - isComplete, [96](#)
 - isValid, [96](#)
- ForIntExpression, [97](#)
 - accept, [99](#)
 - areOperandsValid, [99](#)
 - getTextFormatted, [99](#)
 - getType, [99](#)
 - isComplete, [99](#)
 - isValid, [100](#)
- FunctionCallError, [100](#)
- FunctionCallErrorException, [103](#)
- FunctionCallExpression, [105](#)
 - accept, [106](#)
 - areOperandsValid, [107](#)
 - getArgs, [107](#)
 - getTextFormatted, [107](#)
 - getType, [107](#)
 - hasValidArgs, [108](#)
 - isComplete, [108](#)
 - isValid, [108](#)
- get
 - ErrorCollector< E >, [75](#)
 - Literal, [152](#)
 - StringModifierContainer, [302](#)
- getAll
 - SymTabDuplId< V >, [344](#)
- getAllRules
 - YaraSource, [431](#)
- getAllSemanticErrors
 - YaraSource, [431](#)
- getAllSyntaxErrors
 - YaraSource, [431](#)
- getArg
 - StringModifier, [299](#)
- getArgs
 - FunctionCallExpression, [107](#)
- getBaseDir
 - YaramodConfig, [423](#)
 - YaraSource, [431](#)
- getBuiltinSymbol
 - language.cpp, [475](#)
 - language.h, [451](#)
- getCommentByIndex
 - YaraFile, [402](#)
- getComments
 - YaraFile, [402](#)
- getCondition
 - Rule, [241](#)
- getConst
 - ErrorCollector< E >, [75](#)
- getContent
 - String, [283](#)
- getContext
 - Symbol, [328](#)
- getData
 - Module, [171](#)
- getDataType
 - Module, [171](#)
- getDescription
 - Error, [73](#)
- getEdits
 - YaraFile, [402](#)
- getElementDataType
 - Module, [171](#)
- getElements
 - SetExpression, [275](#)
- getElementType
 - Module, [171](#)
- getEntryFile
 - YaraSource, [432](#)

- getErrorMode
 - YaramodConfig, 423
- getFile
 - FileInclude, 93
 - YaraSource, 432
- getFileIncludes
 - YaraFile, 402
- getFiles
 - YaraSource, 432
- getGlobal
 - RuleModifierContainer, 255
- getId
 - FileInclude, 93
 - Import, 124
 - Meta, 165
 - Rule, 241
 - String, 283
 - Variable, 380
- getIdOffset
 - Rule, 241
- getImport
 - YaraFile, 402, 403
 - YaraSource, 432
- getImports
 - YaraFile, 403
- getInstance
 - ModuleProvider, 179
- getLen
 - YaraFileElement, 414
- getLocalFileInclude
 - YaraFile, 404
- getLocalFileIncludes
 - YaraFile, 404
- getLocalImport
 - YaraFile, 404
- getLocalImports
 - YaraFile, 404
- getLocalOffset
 - RuleElement, 246
- getLocalRule
 - YaraFile, 404
- getLocalRuleByRow
 - YaraFile, 404
- getLocalRules
 - YaraFile, 404
- getMetas
 - Rule, 241
- getMetasById
 - Rule, 241
- getModifiers
 - Rule, 242
 - String, 283
- getModule
 - ModuleProvider, 179
- getModuleDir
 - YaramodConfig, 423
- getName
 - YaraFile, 405
- getOffset
 - RuleElement, 246
 - YaraFileElement, 414
- getParentFile
 - YaraFileElement, 414
- getParentRule
 - RuleElement, 247
- getParsingMode
 - YaramodConfig, 423
- getPath
 - FileInclude, 93
- getPosition
 - TopLevelYaraFileElement, 357
 - YaraFileElement, 414
- getPrivate
 - RuleModifierContainer, 255
- getQuantifier
 - OfExpression, 193
- getRange
 - RuleElement, 247
 - TopLevelYaraFileElement, 357
 - YaraFileElement, 415
- getRaw
 - Literal, 152
- getRule
 - YaraFile, 405
 - YaraSource, 432
- getRules
 - YaraFile, 406
- getSemanticErrors
 - YaraFile, 406
- getSet
 - OfExpression, 193
- getString
 - FileContext, 90
- getStringById
 - Rule, 242
- getStrings
 - Rule, 242
- getSymType
 - Symbol, 328
- getSyntaxErrors
 - YaraFile, 406
- getTags
 - Rule, 242
- getTempString
 - FileContext, 90
- getTextFormatted
 - AllExpression, 20
 - AnyExpression, 24
 - ArrayExpression, 27
 - BinaryExpression, 40
 - Comment, 56
 - Enum, 69
 - ExpressionSetExpression, 87
 - FileInclude, 93
 - ForExpression, 95
 - ForIntExpression, 99

- FunctionCallExpression, [107](#)
- Import, [124](#)
- InExpression, [134](#)
- IntVariable, [138](#)
- Literal, [152](#)
- LiteralExpression, [156](#)
- Meta, [165](#)
- NoneExpression, [186](#)
- OfExpression, [194](#)
- ParenthesesExpression, [204](#)
- PercentQuantifierExpression, [208](#)
- PlainSymbol, [210](#)
- Printable, [215](#)
- Range, [219](#)
- RegexpExpression, [232](#)
- Rule, [242](#)
- RuleSetExpression, [267](#)
- SizeExpression, [278](#)
- String, [283](#)
- StringCountExpression, [287](#)
- StringExpression, [294](#)
- StringMatchLengthExpression, [296](#)
- StringModifier, [299](#)
- StringOffsetExpression, [308](#)
- StringSetExpression, [318](#)
- StructExpression, [323](#)
- Symbol, [328](#)
- ThemExpression, [354](#)
- UnaryExpression, [372](#)
- VarListExpression, [388](#)
- YaraFile, [406](#)
- YaraSource, [433](#)
- getTree
 - FileContext, [90](#)
- getType
 - AtExpression, [29](#)
 - BinaryArithmeticExpression, [32](#)
 - BinaryBitwiseExpression, [34](#)
 - BinaryBoolExpression, [36](#)
 - BinaryRelationalExpression, [42](#)
 - BinaryStrRelationalExpression, [44](#)
 - DefinedExpression, [60](#)
 - Enum, [69](#)
 - Expression, [80](#)
 - ForExpression, [96](#)
 - ForIntExpression, [99](#)
 - FunctionCallExpression, [107](#)
 - InExpression, [135](#)
 - LiteralExpression, [156](#)
 - MatchesExpression, [162](#)
 - OfExpression, [194](#)
 - ParenthesesExpression, [204](#)
 - PercentQuantifierExpression, [208](#)
 - QuantifierExpression, [216](#)
 - Range, [220](#)
 - RegexpExpression, [232](#)
 - RemainderExpression, [235](#)
 - RuleModifier, [253](#)
 - SetExpression, [275](#)
 - SizeExpression, [278](#)
 - String, [283](#)
 - StringCountExpression, [288](#)
 - StringExpression, [294](#)
 - StringMatchLengthExpression, [297](#)
 - StringModifier, [299](#)
 - StringOffsetExpression, [309](#)
 - StructExpression, [323](#)
 - Symbol, [329](#)
 - ThemExpression, [354](#)
 - UnaryArithmeticExpression, [363](#)
 - UnaryBitwiseExpression, [366](#)
 - UnaryBoolExpression, [369](#)
 - VarListExpression, [389](#)
- getValue
 - ExtVariable, [88](#)
 - IntVariable, [138](#)
 - Meta, [165](#)
- getVar
 - YaramodConfig, [423](#)
- getVarById
 - Rule, [242](#)
- getVars
 - Rule, [243](#)
- Global
 - RuleModifier, [253](#)
- GteExpression, [109](#)
 - accept, [110](#)
 - opsign, [110](#)
- GtExpression, [111](#)
 - accept, [112](#)
 - opsign, [112](#)
- has
 - RuleModifierContainer, [255](#)
 - StringModifierContainer, [302](#)
- hasAnyArg
 - StringModifier, [299](#)
- hasArg
 - StringModifier, [299](#)
- hasFastEdit
 - YaraFile, [407](#)
- hasFile
 - YaraSource, [433](#)
- hasGlobal
 - RuleModifierContainer, [255](#)
- hasModule
 - ModuleProvider, [179](#)
- hasPrivate
 - RuleModifierContainer, [256](#)
- hasRule
 - YaraFile, [407](#)
- hasTag
 - Rule, [243](#)
- hasValidArgs
 - FunctionCallExpression, [108](#)
- hasValidCache
 - TopLevelYaraFileElement, [357](#)

- Hex
 - String, 282
- HexStrError, 113
- HexStrErrorException, 115
- IContainsExpression, 117
 - accept, 118
 - opsign, 118
- IEndsWithExpression, 119
 - accept, 120
 - opsign, 120
- IEqExpression, 121
 - accept, 122
 - opsign, 122
- Import, 123
 - getId, 124
 - getTextFormatted, 124
- IncludeError, 125
- IncludeErrorException, 126
- IndexError, 128
- IndexErrorException, 131
- InExpression, 133
 - accept, 134
 - areOperandsValid, 134
 - getTextFormatted, 134
 - getType, 135
 - isComplete, 135
 - isValid, 135
- INITIAL_NODE_STACK_SIZE
 - parser_adapter.h, 460
- INS
 - offset_edit_range_t, 196
- insert
 - SymTab< V >, 341
 - SymTabDuplId< V >, 344
 - SymTabOffset< V >, 347
- InternalErrorException, 136
- IntVariable, 137
 - getTextFormatted, 138
 - getValue, 138
 - setValue, 139
- invalidateCache
 - TopLevelYaraFileElement, 357
- IParserAdapter, 139
 - createYaraSource, 140
 - createYaraSourceFromFile, 140
 - updateYaraSource, 141
 - updateYaraSourceFromFile, 141
- isActive
 - YaraFile, 407
- isAnonymous
 - String, 284
- isArithmeticCompatible
 - Expression, 81
- isComplete
 - ArrayExpression, 27
 - BinaryExpression, 40
 - Enum, 69
 - Expression, 81
- ForExpression, 96
- ForIntExpression, 99
- FunctionCallExpression, 108
- InExpression, 135
- LiteralExpression, 156
- OfExpression, 194
- ParenthesesExpression, 205
- PercentQuantifierExpression, 208
- QuantifierExpression, 217
- RegexExpression, 232
- SetExpression, 275
- SizeExpression, 278
- StringCountExpression, 288
- StringExpression, 294
- StringPropertyExpression, 310
- StructExpression, 323
- Symbol, 329
- ThemExpression, 355
- UnaryExpression, 372
- VarListExpression, 389
- isDefined
 - Symbol, 329
- isDeletionRequested
 - YaraFileElement, 415
- isFixed
 - YaraFileElement, 415
- isGlobal
 - Rule, 243
 - SemanticError, 271
- isIsolated
 - YaraFile, 407
- isPrivate
 - Rule, 243
- IStartsWithExpression, 142
 - accept, 143
 - opsign, 143
- isType
 - Literal, 152
- isUndefined
 - Literal, 153
- isValid
 - ArrayExpression, 27
 - AtExpression, 30
 - BinaryArithmeticExpression, 32
 - BinaryBitwiseExpression, 34
 - BinaryBoolExpression, 36
 - BinaryRelationalExpression, 42
 - BinaryStrRelationalExpression, 44
 - DefinedExpression, 61
 - Enum, 69
 - Expression, 81
 - ForExpression, 96
 - ForIntExpression, 100
 - FunctionCallExpression, 108
 - InExpression, 135
 - MatchesExpression, 163
 - OfExpression, 194
 - ParenthesesExpression, 205

- PercentQuantifierExpression, 208
- QuantifierExpression, 217
- RemainderExpression, 235
- SetExpression, 275
- StringPropertyExpression, 311
- StructExpression, 323
- Symbol, 329
- TerminalExpression, 352
- ThemExpression, 355
- UnaryArithmeticExpression, 363
- UnaryBitwiseExpression, 367
- UnaryBoolExpression, 369
- VarListExpression, 389
- isValidEscapeSequence
 - language.cpp, 476
 - language.h, 451
- isYaraKeyword
 - language.cpp, 476
 - language.h, 451
- jsonToExprType
 - Module, 172
- language.cpp
 - escapeSequenceToChar, 475
 - getBuiltinSymbol, 475
 - isValidEscapeSequence, 476
 - isYaraKeyword, 476
- language.h
 - escapeSequenceToChar, 451
 - getBuiltinSymbol, 451
 - isValidEscapeSequence, 451
 - isYaraKeyword, 451
 - tree_sitter_yara, 452
- LeftShiftExpression, 149
 - accept, 150
 - opsign, 150
- len
 - offset_edit_range_t, 196
- Literal, 151
 - get, 152
 - getRaw, 152
 - getTextFormatted, 152
 - isType, 152
 - isUndefined, 153
 - set, 153
- LiteralExpression, 153
 - accept, 155
 - getTextFormatted, 156
 - getType, 156
 - isComplete, 156
- load
 - Module, 172
- loadModules
 - Yaramod, 418
- loadModulesFromEnvironment
 - Yaramod, 418
- loadModulesFromPath
 - Yaramod, 418
- LteExpression, 157
 - accept, 158
 - opsign, 158
- LtExpression, 159
 - accept, 160
 - opsign, 160
- Manual
 - YaramodConfig, 422
- MatchesExpression, 161
 - accept, 162
 - getType, 162
 - isValid, 163
 - opsign, 163
- merge
 - Module, 172
- mergeJson
 - Module, 172
- Meta, 164
 - getId, 165
 - getTextFormatted, 165
 - getValue, 165
 - setId, 165
 - setValue, 165
- MissingTokenError, 166
- MissingTokenErrorException, 168
- MOD
 - offset_edit_range_t, 196
- modify
 - YaraFile, 407
- Module, 169
 - buildModuleContext, 170
 - findField, 170
 - findInArray, 170
 - getData, 171
 - getDataType, 171
 - getElementDataType, 171
 - getElementType, 171
 - jsonToExprType, 172
 - load, 172
 - merge, 172
 - mergeJson, 172
 - qualifyObject, 172
- ModuleError, 173
- ModuleErrorException, 175
- ModuleProvider, 177
 - addModule, 178
 - cacheSymbol, 178
 - clearCache, 178
 - findCached, 179
 - getInstance, 179
 - getModule, 179
 - hasModule, 179
- moveBindings
 - YaraSource, 433
- MulExpression, 180
 - accept, 181
 - opsign, 181
- mustBeReparsed

- YaraFile, 408
- NeqExpression, 182
 - accept, 183
 - opsign, 183
- node_w_depth_t, 184
- NoneExpression, 184
 - accept, 186
 - getTextFormatted, 186
- normalizeFSPath
 - common.cpp, 436
 - common.h, 440
- NotExpression, 187
 - accept, 188
- notifyAddedSymbol
 - YaraFile, 408
- notifyChange
 - YaraFile, 408
 - YaraSource, 433
- ObservingVisitor, 189
 - start, 191
 - visitBinaryOperatorDefault, 191
- OfExpression, 192
 - accept, 193
 - areOperandsValid, 193
 - getQuantifier, 193
 - getSet, 193
 - getTextFormatted, 194
 - getType, 194
 - isComplete, 194
 - isValid, 194
- offset_edit_range_t, 195
 - DEL, 196
 - EditType, 195
 - INS, 196
 - len, 196
 - MOD, 196
- offset_range_t, 196
- offsetToPoint
 - common.cpp, 436
 - common.h, 440
- opsign
 - AddExpression, 18
 - AndExpression, 22
 - AtExpression, 30
 - BinaryExpression, 40
 - BitwiseAndExpression, 46
 - BitwiseOrExpression, 50
 - BitwiseXorExpression, 52
 - ContainsExpression, 58
 - DivExpression, 63
 - EndsWithExpression, 66
 - EqExpression, 71
 - GteExpression, 110
 - GtExpression, 112
 - IContainsExpression, 118
 - IEndsWithExpression, 120
 - IEqExpression, 122
 - IStartsWithExpression, 143
 - LeftShiftExpression, 150
 - LteExpression, 158
 - LtExpression, 160
 - MatchesExpression, 163
 - MulExpression, 181
 - NeqExpression, 183
 - OrExpression, 198
 - RemainderExpression, 235
 - RightShiftExpression, 237
 - StartsWithExpression, 280
 - SubExpression, 325
- OrExpression, 197
 - accept, 198
 - opsign, 198
- OverflowError, 199
- OverflowErrorException, 200
- ParenthesesExpression, 202
 - accept, 204
 - areOperandsValid, 204
 - getTextFormatted, 204
 - getType, 204
 - isComplete, 205
 - isValid, 205
- parse
 - TSParserAdapter, 360
- parseFile
 - Yaramod, 419
- parser_adapter.h
 - INITIAL_NODE_STACK_SIZE, 460
- parseString
 - Yaramod, 420
- ParsingMode
 - YaramodConfig, 422
- PercentQuantifierExpression, 206
 - accept, 207
 - areOperandsValid, 207
 - getTextFormatted, 208
 - getType, 208
 - isComplete, 208
 - isValid, 208
- Plain
 - String, 282
- PlainSymbol, 209
 - accept, 210
 - getTextFormatted, 210
- point_t, 211
- pointToOffset
 - common.cpp, 436
 - common.h, 440
- Printable, 213
 - getTextFormatted, 215
- Private
 - RuleModifier, 253
- qualifyObject
 - Module, 172
- QuantifierExpression, 215

- areOperandsValid, [216](#)
- getType, [216](#)
- isComplete, [217](#)
- isValid, [217](#)
- Range, [218](#)
 - accept, [219](#)
 - areOperandsValid, [219](#)
 - getTextFormatted, [219](#)
 - getType, [220](#)
- range_t, [220](#)
- RangeError, [221](#)
- RangeErrorException, [224](#)
- Regexp
 - String, [282](#)
- REGEXP_MAXIMUM_QUANTIFIER_NUM
 - string.h, [463](#)
- RegexpError, [226](#)
- RegexpErrorException, [228](#)
- RegexpExpression, [230](#)
 - accept, [231](#)
 - getTextFormatted, [232](#)
 - getType, [232](#)
 - isComplete, [232](#)
- RemainderExpression, [233](#)
 - accept, [234](#)
 - getType, [235](#)
 - isValid, [235](#)
 - opsign, [235](#)
- remove
 - ErrorCollector< E >, [75](#)
 - SymTab< V >, [341](#)
 - SymTabDuplId< V >, [344](#), [345](#)
 - SymTabOffset< V >, [347](#)
- removeComments
 - YaraFile, [408](#)
- removeElementsInRange
 - YaraFile, [408](#)
- removeFile
 - YaraSource, [434](#)
- removeGlobalErrors
 - YaraFile, [408](#)
- removeLocalFileIncludes
 - YaraFile, [409](#)
- removeLocalImports
 - YaraFile, [409](#)
- removeLocalRules
 - YaraFile, [409](#)
- removeMeta
 - Rule, [243](#)
- removeMissingSymbol
 - YaraFile, [409](#)
- removeModifier
 - Rule, [243](#)
- removeSemanticErrors
 - YaraFile, [409](#)
- removeString
 - Rule, [244](#)
- removeSyntaxErrors
 - YaraFile, [409](#)
- removeTag
 - Rule, [244](#)
- removeVar
 - Rule, [244](#)
 - YaramodConfig, [424](#)
- renameFile
 - YaraSource, [434](#)
- reparse
 - TSParserAdapter, [360](#)
- requestDeletion
 - YaraFileElement, [415](#)
- revRemoveUntil
 - ErrorCollector< E >, [75](#)
 - SymTab< V >, [342](#)
 - SymTabDuplId< V >, [345](#)
 - SymTabOffset< V >, [347](#)
- RightShiftExpression, [236](#)
 - accept, [237](#)
 - opsign, [237](#)
- Rule, [238](#)
 - addMeta, [240](#)
 - addModifier, [240](#)
 - addString, [240](#)
 - addTag, [240](#)
 - addVar, [241](#)
 - getCondition, [241](#)
 - getId, [241](#)
 - getIdOffset, [241](#)
 - getMetas, [241](#)
 - getMetasById, [241](#)
 - getModifiers, [242](#)
 - getStringById, [242](#)
 - getStrings, [242](#)
 - getTags, [242](#)
 - getTextFormatted, [242](#)
 - getVarById, [242](#)
 - getVars, [243](#)
 - hasTag, [243](#)
 - isGlobal, [243](#)
 - isPrivate, [243](#)
 - removeMeta, [243](#)
 - removeModifier, [243](#)
 - removeString, [244](#)
 - removeTag, [244](#)
 - removeVar, [244](#)
 - setCondition, [244](#)
 - setId, [244](#)
 - setIdOffset, [244](#)
- RuleElement, [245](#)
 - getLocalOffset, [246](#)
 - getOffset, [246](#)
 - getParentRule, [247](#)
 - getRange, [247](#)
 - setLocalOffset, [247](#)
 - setOffset, [247](#)
 - setParentRule, [247](#)
- RuleError, [248](#)

- RuleErrorException, 249
- ruleLookup
 - YaraFile, 410
- RuleModifier, 251
 - getType, 253
 - Global, 253
 - Private, 253
 - stringToType, 253
 - Type, 253
 - typeToString, 253
- RuleModifierContainer, 254
 - add, 254
 - begin, 254
 - empty, 254
 - end, 255
 - erase, 255
 - getGlobal, 255
 - getPrivate, 255
 - has, 255
 - hasGlobal, 255
 - hasPrivate, 256
 - size, 256
- RuleModifierContainer::iterator, 144
- RuleModifierError, 256
- RuleModifierErrorException, 259
- RuleReferenceError, 261
- RuleReferenceErrorException, 263
- RuleSetExpression, 265
 - accept, 266
 - getTextFormatted, 267
- RuleWildcardExpression, 268
 - accept, 269
- search
 - SymTab< V >, 342
- SemanticError, 270
 - isGlobal, 271
 - setGlobal, 271
- SemanticErrorException, 272
- set
 - Literal, 153
- setActivity
 - YaraFile, 410
- setAnonymous
 - String, 284
- setArg
 - StringModifier, 299
- setBaseDir
 - YaramodConfig, 424
- setCondition
 - Rule, 244
- setContent
 - String, 284
- setContext
 - Symbol, 330
- setDefined
 - Symbol, 330
- setEntryFile
 - YaraSource, 434
- setErrorMode
 - YaraFile, 410
 - YaramodConfig, 424
- SetExpression, 273
 - areOperandsValid, 274
 - getElements, 275
 - getType, 275
 - isComplete, 275
 - isValid, 275
- setGlobal
 - SemanticError, 271
- setId
 - Meta, 165
 - Rule, 244
 - String, 284
 - Variable, 380
- setIdOffset
 - Rule, 244
- setIsolated
 - YaraFile, 410
- setLen
 - YaraFileElement, 415
- setLocalOffset
 - RuleElement, 247
- setModuleDir
 - YaramodConfig, 424
- setName
 - YaraFile, 410
- setOffset
 - RuleElement, 247
 - YaraFileElement, 415
- setParent
 - YaraFileElement, 415
- setParentFile
 - YaraFileElement, 416
- setParentRule
 - RuleElement, 247
- setParsingMode
 - YaramodConfig, 424
- setRangeCache
 - TopLevelYaraFileElement, 357
- setString
 - FileContext, 90
- setSymType
 - Symbol, 330
- setTree
 - FileContext, 90
- setValue
 - ExtVariable, 88
 - IntVariable, 139
 - Meta, 165
- shiftElements
 - YaraFile, 411
- size
 - RuleModifierContainer, 256
 - StringModifierContainer, 302
- SizeExpression, 276
 - accept, 277

- getFormatted, 278
- getType, 278
- isComplete, 278
- unitToString, 278
- start
 - ObservingVisitor, 191
- StartsWithExpression, 279
 - accept, 280
 - opsign, 280
- Strict
 - YaramodConfig, 422
- String, 281
 - addModifier, 283
 - getContent, 283
 - getId, 283
 - getModifiers, 283
 - getFormatted, 283
 - getType, 283
 - Hex, 282
 - isAnonymous, 284
 - Plain, 282
 - Regex, 282
 - setAnonymous, 284
 - setContent, 284
 - setId, 284
 - Type, 282
 - typeToString, 284
- string.h
 - BASE64_ALPHA_LEN, 463
 - REGEXP_MAXIMUM_QUANTIFIER_NUM, 463
 - XOR_UPPER_RANGE_BOUND, 464
- string_hash_t, 285
- StringCountExpression, 286
 - accept, 287
 - getFormatted, 287
 - getType, 288
 - isComplete, 288
- stringEdit
 - FileContext, 90
- StringError, 289
- StringErrorException, 290
- StringExpression, 292
 - accept, 293
 - getFormatted, 294
 - getType, 294
 - isComplete, 294
- StringMatchLengthExpression, 295
 - accept, 296
 - getFormatted, 296
 - getType, 297
- StringModifier, 297
 - getArg, 299
 - getFormatted, 299
 - getType, 299
 - hasAnyArg, 299
 - hasArg, 299
 - setArg, 299
 - stringToType, 300
 - typeToString, 300
- StringModifierContainer, 300
 - add, 301
 - begin, 301
 - empty, 301
 - end, 301
 - erase, 301
 - get, 302
 - has, 302
 - size, 302
- StringModifierContainer::iterator, 145
- StringModifierError, 303
- StringModifierErrorException, 305
- StringOffsetExpression, 307
 - accept, 308
 - getFormatted, 308
 - getType, 309
- StringPropertyExpression, 309
 - areOperandsValid, 310
 - isComplete, 310
 - isValid, 311
- StringReferenceError, 311
- StringReferenceErrorException, 314
- StringSetExpression, 316
 - accept, 317
 - getFormatted, 318
- stringToType
 - RuleModifier, 253
 - StringModifier, 300
- stringUpdate
 - FileContext, 91
- StringWildcardExpression, 319
 - accept, 320
- StructExpression, 321
 - accept, 322
 - areOperandsValid, 323
 - getFormatted, 323
 - getType, 323
 - isComplete, 323
 - isValid, 323
- stype_
 - Symbol, 331
- SubExpression, 324
 - accept, 325
 - opsign, 325
- Symbol, 326
 - areOperandsValid, 328
 - ctx_, 330
 - defined_, 330
 - dtype_, 331
 - getContext, 328
 - getSymType, 328
 - getFormatted, 328
 - getType, 329
 - isComplete, 329
 - isDefined, 329
 - isValid, 329
 - setContext, 330

- setDefined, 330
 - setSymType, 330
 - stype_, 331
- SymbolError, 331
- SymbolErrorException, 334
- SymbolReferenceError, 336
- SymbolReferenceErrorException, 338
- SymTab< V >, 339
 - clear, 340
 - dataById, 341
 - dataByOffset, 341
 - insert, 341
 - remove, 341
 - revRemoveUntil, 342
 - search, 342
 - update, 342
- SymTab< V >::iterator, 145
- SymTabDuplId< V >, 343
 - at, 344
 - clear, 344
 - exists, 344
 - getAll, 344
 - insert, 344
 - remove, 344, 345
 - revRemoveUntil, 345
 - update, 345
- SymTabDuplId< V >::iterator, 146
- SymTabDuplId< V >::Pool, 212
 - erase, 212
- SymTabDuplId< V >::Pool::iterator, 147
- SymTabOffset< V >, 346
 - exists, 347
 - insert, 347
 - remove, 347
 - revRemoveUntil, 347
 - update, 347
- SymTabOffset< V >::iterator, 148
- SyntaxError, 348
- SyntaxErrorException, 349
- TerminalExpression, 351
 - areOperandsValid, 352
 - isValid, 352
- ThemExpression, 353
 - accept, 354
 - areOperandsValid, 354
 - getTextFormatted, 354
 - getType, 354
 - isComplete, 355
 - isValid, 355
- Tolerant
 - YaramodConfig, 422
- TopLevelYaraFileElement, 355
 - getPosition, 357
 - getRange, 357
 - hasValidCache, 357
 - invalidateCache, 357
 - setRangeCache, 357
- toTSEdit
 - FileContext, 91
- tree_sitter_yara
 - language.h, 452
- treeEdit
 - FileContext, 91
- TSParserAdapter, 358
 - createYaraSource, 359
 - createYaraSourceFromFile, 359
 - parse, 360
 - reparse, 360
 - updateYaraSource, 360
 - updateYaraSourceFromFile, 361
- Type
 - RuleModifier, 253
 - String, 282
- typeToString
 - Expression, 82
 - RuleModifier, 253
 - String, 284
 - StringModifier, 300
- UnaryArithmeticExpression, 361
 - getType, 363
 - isValid, 363
- UnaryBitwiseExpression, 364
 - getType, 366
 - isValid, 367
- UnaryBoolExpression, 367
 - getType, 369
 - isValid, 369
- UnaryExpression, 370
 - areOperandsValid, 371
 - getTextFormatted, 372
 - isComplete, 372
- UnaryMinusExpression, 373
 - accept, 374
- undo
 - YaraFile, 411
 - YaraSource, 434
- undoEdits
 - FileContext, 91
- UnexpectedTokenError, 375
- UnexpectedTokenErrorException, 376
- unitToString
 - SizeExpression, 278
- update
 - ErrorCollector< E >, 75
 - SymTab< V >, 342
 - SymTabDuplId< V >, 345
 - SymTabOffset< V >, 347
- updateComments
 - YaraFile, 411
- updateLocalFileIncludes
 - YaraFile, 411
- updateLocalImports
 - YaraFile, 411
- updateLocalRules
 - YaraFile, 411
- updateModifiedRanges

- yara_file.cpp, 486
- updateSyntaxErrors
 - YaraFile, 412
- updateYaraSource
 - IParserAdapter, 141
 - TSParserAdapter, 360
- updateYaraSourceFromFile
 - IParserAdapter, 141
 - TSParserAdapter, 361
- var_def_t, 378
- Variable, 379
 - getId, 380
 - setId, 380
- VariableError, 380
- VariableErrorException, 382
- VariableExpression, 385
 - accept, 386
- VarListExpression, 387
 - accept, 388
 - areOperandsValid, 388
 - getTextFormatted, 388
 - getType, 389
 - isComplete, 389
 - isValid, 389
- visitBinaryOperatorDefault
 - ObservingVisitor, 191
- Visitor, 390
- wasEdited
 - YaraFile, 412
- XOR_UPPER_RANGE_BOUND
 - string.h, 464
- yara_file.cpp
 - updateModifiedRanges, 486
- YaraErrorException, 392
- YaraFile, 393
 - addComment, 396
 - addEdit, 396
 - addError, 396
 - addFileInclude, 396
 - addImport, 397
 - addMissingSymbol, 397
 - addRule, 397
 - addSemanticError, 397
 - addSyntaxError, 397
 - addWrong, 398
 - bind, 398
 - checkDuplicatedRules, 398
 - checkMissing, 398
 - checkRange, 399
 - clearAll, 399
 - clearComments, 399
 - clearEdits, 399
 - clearLocalFileIncludes, 399
 - clearLocalImports, 399
 - clearLocalRules, 399
 - clearSemanticErrors, 400
 - clearSyntaxErrors, 400
 - clearToBeChecked, 400
 - clearWrong, 400
 - createRuleError, 400
 - ctx, 400
 - deleteBindings, 400
 - dumpSyntaxErrors, 401
 - edit, 401
 - getCommentByIndex, 402
 - getComments, 402
 - getEdits, 402
 - getFileIncludes, 402
 - getImport, 402, 403
 - getImports, 403
 - getLocalFileInclude, 404
 - getLocalFileIncludes, 404
 - getLocalImport, 404
 - getLocalImports, 404
 - getLocalRule, 404
 - getLocalRuleByRow, 404
 - getLocalRules, 404
 - getName, 405
 - getRule, 405
 - getRules, 406
 - getSemanticErrors, 406
 - getSyntaxErrors, 406
 - getTextFormatted, 406
 - hasFastEdit, 407
 - hasRule, 407
 - isActive, 407
 - isIsolated, 407
 - modify, 407
 - mustBeReparsed, 408
 - notifyAddedSymbol, 408
 - notifyChange, 408
 - removeComments, 408
 - removeElementsInRange, 408
 - removeGlobalErrors, 408
 - removeLocalFileIncludes, 409
 - removeLocalImports, 409
 - removeLocalRules, 409
 - removeMissingSymbol, 409
 - removeSemanticErrors, 409
 - removeSyntaxErrors, 409
 - ruleLookup, 410
 - setActivity, 410
 - setErrorMode, 410
 - setIsolated, 410
 - setName, 410
 - shiftElements, 411
 - undo, 411
 - updateComments, 411
 - updateLocalFileIncludes, 411
 - updateLocalImports, 411
 - updateLocalRules, 411
 - updateSyntaxErrors, 412
 - wasEdited, 412

- YaraFileElement, 412
 - getLen, 414
 - getOffset, 414
 - getParentFile, 414
 - getPosition, 414
 - getRange, 415
 - isDeletionRequested, 415
 - isFixed, 415
 - requestDeletion, 415
 - setLen, 415
 - setOffset, 415
 - setParent, 415
 - setParentFile, 416
- YaraFileElementBindable, 416
 - bind, 417
 - bind_to, 417
- Yaramod, 417
 - config, 418
 - loadModules, 418
 - loadModulesFromEnvironment, 418
 - loadModulesFromPath, 418
 - parseFile, 419
 - parseString, 420
- YaramodConfig, 421
 - addBoolVar, 422
 - addVar, 423
 - Auto, 422
 - ErrorMode, 421
 - getBaseDir, 423
 - getErrorMode, 423
 - getModuleDir, 423
 - getParsingMode, 423
 - getVar, 423
 - Manual, 422
 - ParsingMode, 422
 - removeVar, 424
 - setBaseDir, 424
 - setErrorMode, 424
 - setModuleDir, 424
 - setParsingMode, 424
 - Strict, 422
 - Tolerant, 422
 - YaramodConfig, 422
- YaramodErrorException, 425
- YaraSource, 426
 - addFile, 428
 - bind, 428
 - clearInactiveFiles, 428
 - deactivateFiles, 428
 - deleteBindings, 428
 - dumpBindChildren, 428
 - dumpBindParents, 429
 - edit, 429, 430
 - getAllRules, 431
 - getAllSemanticErrors, 431
 - getAllSyntaxErrors, 431
 - getBaseDir, 431
 - getEntryFile, 432
 - getFile, 432
 - getFiles, 432
 - getImport, 432
 - getRule, 432
 - getTextFormatted, 433
 - hasFile, 433
 - moveBindings, 433
 - notifyChange, 433
 - removeFile, 434
 - renameFile, 434
 - setEntryFile, 434
 - undo, 434