

ebnf2tikz

Generated by Doxygen 1.9.1

1 Contributor Covenant Code of Conduct	1
1.1 Our Pledge	1
1.2 Our Standards	1
1.3 Our Responsibilities	2
1.4 Scope	2
1.5 Enforcement	2
1.6 Attribution	2
2 CONTRIBUTING	3
3 ebnf2tikz	5
3.1 What Does It Do?	5
3.2 About the Code	6
3.3 To Do	7
4 Namespace Index	9
4.1 Namespace List	9
5 Hierarchical Index	11
5.1 Class Hierarchy	11
6 Class Index	13
6.1 Class List	13
7 File Index	15
7.1 File List	15
8 Namespace Documentation	17
8.1 annot Namespace Reference	17
8.1.1 Function Documentation	18
8.1.1.1 operator+() [1/3]	18
8.1.1.2 operator+() [2/3]	18
8.1.1.3 operator+() [3/3]	18
8.1.1.4 operator+=() [1/3]	18
8.1.1.5 operator+=() [2/3]	18
8.1.1.6 operator+=() [3/3]	19
8.1.1.7 operator-() [1/2]	19
8.1.1.8 operator-() [2/2]	19
8.1.1.9 operator-=() [1/2]	19
8.1.1.10 operator-=() [2/2]	19
8.1.1.11 operator<<() [1/2]	19
8.1.1.12 operator<<() [2/2]	20
8.2 yy Namespace Reference	20
8.2.1 Function Documentation	21
8.2.1.1 operator+() [1/3]	21

8.2.1.2 operator+() [2/3]	21
8.2.1.3 operator+() [3/3]	21
8.2.1.4 operator+=() [1/3]	22
8.2.1.5 operator+=() [2/3]	22
8.2.1.6 operator+=() [3/3]	22
8.2.1.7 operator-() [1/2]	22
8.2.1.8 operator-() [2/2]	22
8.2.1.9 operator-=() [1/2]	23
8.2.1.10 operator-=() [2/2]	23
8.2.1.11 operator<<() [1/2]	23
8.2.1.12 operator<<() [2/2]	23
9 Class Documentation	25
9.1 annot::parser::basic_symbol< Base > Struct Template Reference	25
9.1.1 Detailed Description	27
9.1.2 Member Typedef Documentation	27
9.1.2.1 super_type	27
9.1.3 Constructor & Destructor Documentation	27
9.1.3.1 basic_symbol() [1/6]	27
9.1.3.2 basic_symbol() [2/6]	28
9.1.3.3 basic_symbol() [3/6]	28
9.1.3.4 basic_symbol() [4/6]	28
9.1.3.5 basic_symbol() [5/6]	28
9.1.3.6 basic_symbol() [6/6]	28
9.1.3.7 ~basic_symbol()	29
9.1.4 Member Function Documentation	29
9.1.4.1 clear()	29
9.1.4.2 empty()	29
9.1.4.3 move()	29
9.1.4.4 name()	29
9.1.4.5 type_get()	30
9.1.5 Member Data Documentation	30
9.1.5.1 location	30
9.1.5.2 value	30
9.2 yy::parser::basic_symbol< Base > Struct Template Reference	30
9.2.1 Detailed Description	32
9.2.2 Member Typedef Documentation	32
9.2.2.1 super_type	32
9.2.3 Constructor & Destructor Documentation	32
9.2.3.1 basic_symbol() [1/8]	32
9.2.3.2 basic_symbol() [2/8]	33
9.2.3.3 basic_symbol() [3/8]	33

9.2.3.4 basic_symbol() [4/8]	33
9.2.3.5 basic_symbol() [5/8]	33
9.2.3.6 basic_symbol() [6/8]	33
9.2.3.7 basic_symbol() [7/8]	34
9.2.3.8 basic_symbol() [8/8]	34
9.2.3.9 ~basic_symbol()	34
9.2.4 Member Function Documentation	34
9.2.4.1 clear()	34
9.2.4.2 empty()	34
9.2.4.3 move()	35
9.2.4.4 name()	35
9.2.4.5 type_get()	35
9.2.5 Member Data Documentation	35
9.2.5.1 location	35
9.2.5.2 value	35
9.3 annot::parser::by_kind Struct Reference	36
9.3.1 Detailed Description	37
9.3.2 Member Typedef Documentation	37
9.3.2.1 kind_type	37
9.3.3 Constructor & Destructor Documentation	37
9.3.3.1 by_kind() [1/3]	37
9.3.3.2 by_kind() [2/3]	37
9.3.3.3 by_kind() [3/3]	37
9.3.4 Member Function Documentation	37
9.3.4.1 clear()	38
9.3.4.2 kind()	38
9.3.4.3 move()	38
9.3.4.4 type_get()	38
9.3.5 Member Data Documentation	38
9.3.5.1 kind_	38
9.4 yy::parser::by_kind Struct Reference	38
9.4.1 Detailed Description	39
9.4.2 Member Typedef Documentation	39
9.4.2.1 kind_type	39
9.4.3 Constructor & Destructor Documentation	39
9.4.3.1 by_kind() [1/3]	40
9.4.3.2 by_kind() [2/3]	40
9.4.3.3 by_kind() [3/3]	40
9.4.4 Member Function Documentation	40
9.4.4.1 clear()	40
9.4.4.2 kind()	40
9.4.4.3 move()	40

9.4.4.4 type_get()	41
9.4.5 Member Data Documentation	41
9.4.5.1 kind_	41
9.5 choicenode Class Reference	41
9.5.1 Constructor & Destructor Documentation	42
9.5.1.1 choicenode() [1/2]	43
9.5.1.2 choicenode() [2/2]	43
9.5.1.3 ~choicenode()	43
9.5.2 Member Function Documentation	43
9.5.2.1 clone()	43
9.5.2.2 drawToLeftRail()	43
9.5.2.3 drawToRightRail()	44
9.5.2.4 dump()	44
9.5.2.5 fixSkips()	44
9.5.2.6 insert()	44
9.5.2.7 mergeChoices()	44
9.5.2.8 rail_left()	45
9.5.2.9 rail_right()	45
9.5.2.10 texName()	45
9.6 concatnode Class Reference	45
9.6.1 Constructor & Destructor Documentation	46
9.6.1.1 concatnode() [1/2]	47
9.6.1.2 concatnode() [2/2]	47
9.6.1.3 ~concatnode()	47
9.6.2 Member Function Documentation	47
9.6.2.1 analyzeNonOptLoops()	47
9.6.2.2 analyzeOptLoops()	47
9.6.2.3 clone()	47
9.6.2.4 createRows()	48
9.6.2.5 drawToLeftRail()	48
9.6.2.6 drawToRightRail()	48
9.6.2.7 dump()	48
9.6.2.8 fixSkips()	48
9.6.2.9 insert()	49
9.6.2.10 mergeConcats()	49
9.6.2.11 mergeRails()	49
9.6.2.12 place()	49
9.6.2.13 setNext()	49
9.6.2.14 setPrevious()	50
9.7 annot::parser::context Class Reference	50
9.7.1 Constructor & Destructor Documentation	50
9.7.1.1 context()	50

9.7.2 Member Function Documentation	50
9.7.2.1 expected_tokens()	50
9.7.2.2 location()	51
9.7.2.3 lookahead()	51
9.7.2.4 token()	51
9.8 yy::parser::context Class Reference	51
9.8.1 Constructor & Destructor Documentation	51
9.8.1.1 context()	51
9.8.2 Member Function Documentation	52
9.8.2.1 expected_tokens()	52
9.8.2.2 location()	52
9.8.2.3 lookahead()	52
9.8.2.4 token()	52
9.9 coordinate Class Reference	52
9.9.1 Constructor & Destructor Documentation	53
9.9.1.1 coordinate() [1/2]	53
9.9.1.2 coordinate() [2/2]	53
9.9.2 Member Function Documentation	53
9.9.2.1 operator+()	53
9.9.2.2 operator-()	53
9.9.2.3 operator=()	54
9.9.3 Friends And Related Function Documentation	54
9.9.3.1 operator<<	54
9.9.4 Member Data Documentation	54
9.9.4.1 x	54
9.9.4.2 y	54
9.10 driver Class Reference	54
9.10.1 Constructor & Destructor Documentation	55
9.10.1.1 driver()	55
9.10.2 Member Function Documentation	55
9.10.2.1 addString()	55
9.10.2.2 addTerminal()	55
9.10.2.3 get_location()	55
9.10.2.4 get_result()	55
9.10.2.5 outs()	55
9.10.2.6 parse()	56
9.10.2.7 scan_begin() [1/2]	56
9.10.2.8 scan_begin() [2/2]	56
9.10.2.9 scan_end()	56
9.11 grammar Class Reference	56
9.11.1 Constructor & Destructor Documentation	57
9.11.1.1 grammar()	57

9.11.1.2 ~grammar()	57
9.11.2 Member Function Documentation	57
9.11.2.1 createRows()	57
9.11.2.2 dump()	57
9.11.2.3 fixSkips()	57
9.11.2.4 insert()	57
9.11.2.5 mergeRails()	58
9.11.2.6 optimize()	58
9.11.2.7 place()	58
9.11.2.8 setNext()	58
9.11.2.9 setParent()	58
9.11.2.10 setPrevious()	58
9.11.2.11 subsume()	58
9.12 annot::location Class Reference	59
9.12.1 Detailed Description	60
9.12.2 Member Typedef Documentation	60
9.12.2.1 counter_type	60
9.12.2.2 filename_type	60
9.12.3 Constructor & Destructor Documentation	60
9.12.3.1 location() [1/3]	60
9.12.3.2 location() [2/3]	61
9.12.3.3 location() [3/3]	61
9.12.4 Member Function Documentation	61
9.12.4.1 columns()	61
9.12.4.2 initialize()	61
9.12.4.3 lines()	61
9.12.4.4 step()	62
9.12.5 Member Data Documentation	62
9.12.5.1 begin	62
9.12.5.2 end	62
9.13 yy::location Class Reference	62
9.13.1 Detailed Description	63
9.13.2 Member Typedef Documentation	63
9.13.2.1 counter_type	63
9.13.2.2 filename_type	64
9.13.3 Constructor & Destructor Documentation	64
9.13.3.1 location() [1/3]	64
9.13.3.2 location() [2/3]	64
9.13.3.3 location() [3/3]	64
9.13.4 Member Function Documentation	64
9.13.4.1 columns()	64
9.13.4.2 initialize()	65

9.13.4.3 lines()	65
9.13.4.4 step()	65
9.13.5 Member Data Documentation	65
9.13.5.1 begin	65
9.13.5.2 end	65
9.14 loopnode Class Reference	66
9.14.1 Constructor & Destructor Documentation	67
9.14.1.1 loopnode() [1/2]	67
9.14.1.2 loopnode() [2/2]	67
9.14.1.3 ~loopnode()	67
9.14.2 Member Function Documentation	67
9.14.2.1 clone()	68
9.14.2.2 drawToLeftRail()	68
9.14.2.3 drawToRightRail()	68
9.14.2.4 dump()	68
9.14.2.5 fixSkips()	68
9.14.2.6 getBody()	69
9.14.2.7 getRepeat()	69
9.14.2.8 setBody()	69
9.14.2.9 setRepeat()	69
9.14.2.10 texName()	69
9.15 multinode Class Reference	70
9.15.1 Constructor & Destructor Documentation	71
9.15.1.1 multinode() [1/2]	71
9.15.1.2 multinode() [2/2]	71
9.15.1.3 ~multinode()	72
9.15.2 Member Function Documentation	72
9.15.2.1 analyzeNonOptLoops()	72
9.15.2.2 analyzeOptLoops()	72
9.15.2.3 clone()	72
9.15.2.4 fixSkips()	72
9.15.2.5 forgetChild()	73
9.15.2.6 getChild()	73
9.15.2.7 insert()	73
9.15.2.8 insertFirst()	73
9.15.2.9 liftConcats()	73
9.15.2.10 mergeChoices()	74
9.15.2.11 mergeConcats()	74
9.15.2.12 mergeRails()	74
9.15.2.13 numChildren()	74
9.15.2.14 operator"!="()	74
9.15.2.15 operator=="()	75

9.15.2.16 place()	75
9.15.2.17 setNext()	75
9.15.2.18 setParent()	75
9.15.2.19 setPrevious()	75
9.15.2.20 subsume()	76
9.15.2.21 texName()	76
9.15.3 Friends And Related Function Documentation	76
9.15.3.1 concatnode	76
9.15.4 Member Data Documentation	76
9.15.4.1 nodes	76
9.16 newlinenode Class Reference	77
9.16.1 Constructor & Destructor Documentation	78
9.16.1.1 newlinenode() [1/2]	78
9.16.1.2 newlinenode() [2/2]	78
9.16.1.3 ~newlinenode()	78
9.16.2 Member Function Documentation	78
9.16.2.1 clone()	78
9.16.2.2 dump()	79
9.16.2.3 place()	79
9.16.2.4 rail_left()	79
9.16.2.5 rail_right()	79
9.16.2.6 setLineHeight()	79
9.17 node Class Reference	80
9.17.1 Member Enumeration Documentation	82
9.17.1.1 nodetype	82
9.17.2 Constructor & Destructor Documentation	83
9.17.2.1 node() [1/2]	83
9.17.2.2 node() [2/2]	83
9.17.2.3 ~node()	83
9.17.3 Member Function Documentation	83
9.17.3.1 analyzeNonOptLoops()	83
9.17.3.2 analyzeOptLoops()	84
9.17.3.3 clone()	84
9.17.3.4 createRows()	84
9.17.3.5 deleteData()	84
9.17.3.6 drawToLeftRail()	84
9.17.3.7 drawToRightRail()	85
9.17.3.8 dump()	85
9.17.3.9 east()	85
9.17.3.10 fixSkips()	85
9.17.3.11 forgetChild()	85
9.17.3.12 getBeforeSkip()	85

9.17.3.13 getChild()	86
9.17.3.14 getColSep()	86
9.17.3.15 getDrawToPrev()	86
9.17.3.16 getLeftRail()	86
9.17.3.17 getNext()	86
9.17.3.18 getParent()	86
9.17.3.19 getPrevious()	86
9.17.3.20 getRightRail()	87
9.17.3.21 height()	87
9.17.3.22 insert()	87
9.17.3.23 is_choice()	87
9.17.3.24 is_concat()	87
9.17.3.25 is_loop()	87
9.17.3.26 is_newline()	87
9.17.3.27 is_nonterm()	88
9.17.3.28 is_null()	88
9.17.3.29 is_production()	88
9.17.3.30 is_rail()	88
9.17.3.31 is_row()	88
9.17.3.32 is_terminal()	88
9.17.3.33 isDead()	88
9.17.3.34 liftConcats()	89
9.17.3.35 line()	89
9.17.3.36 loadData()	89
9.17.3.37 makeDead()	89
9.17.3.38 mergeChoices()	89
9.17.3.39 mergeConcats()	89
9.17.3.40 mergeRails()	90
9.17.3.41 numChildren()	90
9.17.3.42 operator"!=()	90
9.17.3.43 operator=="()	90
9.17.3.44 place()	90
9.17.3.45 rawName()	91
9.17.3.46 same_type()	91
9.17.3.47 setBeforeSkip()	91
9.17.3.48 setDrawToPrev()	91
9.17.3.49 setheight()	91
9.17.3.50 setLeftRail()	91
9.17.3.51 setNext()	91
9.17.3.52 setParent()	92
9.17.3.53 setPrevious()	92
9.17.3.54 setRightRail()	92

9.17.3.55 setwidth()	92
9.17.3.56 subsume()	92
9.17.3.57 texName()	92
9.17.3.58 vtrailStr()	93
9.17.3.59 west()	93
9.17.3.60 width()	93
9.17.4 Member Data Documentation	93
9.17.4.1 beforeskip	93
9.17.4.2 dead	93
9.17.4.3 drawtoprev	93
9.17.4.4 ea	93
9.17.4.5 leftrail	94
9.17.4.6 location	94
9.17.4.7 myHeight	94
9.17.4.8 myWidth	94
9.17.4.9 next	94
9.17.4.10 nodename	94
9.17.4.11 parent	94
9.17.4.12 previous	94
9.17.4.13 rightrail	95
9.17.4.14 sizes	95
9.17.4.15 type	95
9.17.4.16 wa	95
9.18 nodesizes Class Reference	95
9.18.1 Constructor & Destructor Documentation	96
9.18.1.1 nodesizes()	96
9.18.1.2 ~nodesizes()	96
9.18.2 Member Function Documentation	96
9.18.2.1 getSize()	96
9.18.2.2 loadData()	96
9.18.3 Member Data Documentation	96
9.18.3.1 colsep	96
9.18.3.2 minsize	97
9.18.3.3 rowsep	97
9.19 nontermnode Class Reference	97
9.19.1 Constructor & Destructor Documentation	99
9.19.1.1 nontermnode() [1/2]	99
9.19.1.2 nontermnode() [2/2]	99
9.19.1.3 ~nontermnode()	99
9.19.2 Member Function Documentation	99
9.19.2.1 analyzeNonOptLoops()	99
9.19.2.2 analyzeOptLoops()	99

9.19.2.3 clone()	100
9.19.2.4 drawToLeftRail()	100
9.19.2.5 drawToRightRail()	100
9.19.2.6 dump()	100
9.19.2.7 forgetChild()	100
9.19.2.8 getChild()	101
9.19.2.9 liftConcats()	101
9.19.2.10 mergeChoices()	101
9.19.2.11 mergeConcats()	101
9.19.2.12 operator"!=()	101
9.19.2.13 operator==()	101
9.19.2.14 place()	102
9.19.2.15 subsume()	102
9.19.2.16 texName()	102
9.19.3 Member Data Documentation	102
9.19.3.1 format	102
9.19.3.2 str	102
9.19.3.3 style	103
9.20 nullnode Class Reference	103
9.20.1 Constructor & Destructor Documentation	104
9.20.1.1 nullnode() [1/2]	104
9.20.1.2 nullnode() [2/2]	105
9.20.2 Member Function Documentation	105
9.20.2.1 clone()	105
9.20.2.2 place()	105
9.20.2.3 texName()	105
9.21 annot::parser Class Reference	105
9.21.1 Detailed Description	107
9.21.2 Member Typedef Documentation	107
9.21.2.1 by_type	107
9.21.2.2 location_type	107
9.21.2.3 symbol_kind_type	107
9.21.2.4 token_kind_type	107
9.21.2.5 token_type	108
9.21.3 Constructor & Destructor Documentation	108
9.21.3.1 parser()	108
9.21.3.2 ~parser()	108
9.21.4 Member Function Documentation	108
9.21.4.1 error() [1/2]	108
9.21.4.2 error() [2/2]	108
9.21.4.3 make_AEND()	109
9.21.4.4 make_ANNOTerror()	109

9.21.4.5 make_ANNOTUNDEF()	109
9.21.4.6 make_AS()	109
9.21.4.7 make_ASTART()	109
9.21.4.8 make_CAPTION()	109
9.21.4.9 make_END()	110
9.21.4.10 make_SEMICOLON()	110
9.21.4.11 make_SIDEWAYS()	110
9.21.4.12 make_STRING()	110
9.21.4.13 make_SUBSUME()	110
9.21.4.14 make_UNEXP()	110
9.21.4.15 operator()	111
9.21.4.16 parse()	111
9.21.4.17 symbol_name()	111
9.21.5 Member Data Documentation	111
9.21.5.1 YYTOKENS	111
9.22 yy::parser Class Reference	112
9.22.1 Detailed Description	113
9.22.2 Member Typedef Documentation	114
9.22.2.1 by_type	114
9.22.2.2 debug_level_type	114
9.22.2.3 location_type	114
9.22.2.4 semantic_type	114
9.22.2.5 symbol_kind_type	114
9.22.2.6 token_kind_type	115
9.22.2.7 token_type	115
9.22.3 Constructor & Destructor Documentation	115
9.22.3.1 parser()	115
9.22.3.2 ~parser()	115
9.22.4 Member Function Documentation	115
9.22.4.1 debug_level()	115
9.22.4.2 debug_stream()	115
9.22.4.3 error() [1/2]	115
9.22.4.4 error() [2/2]	116
9.22.4.5 make_ANNOTATION()	116
9.22.4.6 make_COMMA()	116
9.22.4.7 make_END()	116
9.22.4.8 make_EQUAL()	116
9.22.4.9 make_LBRACE()	117
9.22.4.10 make_LBRACK()	117
9.22.4.11 make_LPAREN()	117
9.22.4.12 make_NEWLINE()	117
9.22.4.13 make_PIPE()	117

9.22.4.14	make_RBRACE()	117
9.22.4.15	make_RBRACK()	117
9.22.4.16	make_RPAREN()	118
9.22.4.17	make_SEMICOLON()	118
9.22.4.18	make_STRING()	118
9.22.4.19	make_TERM()	118
9.22.4.20	make_UNEXP()	118
9.22.4.21	make_YYerror()	118
9.22.4.22	make_YYUNDEF()	119
9.22.4.23	operator()	119
9.22.4.24	parse()	119
9.22.4.25	set_debug_level()	119
9.22.4.26	set_debug_stream()	120
9.22.4.27	symbol_name()	120
9.22.5	Member Data Documentation	120
9.22.5.1	YYTOKENS	120
9.23	annot::position Class Reference	120
9.23.1	Detailed Description	121
9.23.2	Member Typedef Documentation	121
9.23.2.1	counter_type	121
9.23.2.2	filename_type	121
9.23.3	Constructor & Destructor Documentation	122
9.23.3.1	position()	122
9.23.4	Member Function Documentation	122
9.23.4.1	columns()	122
9.23.4.2	initialize()	122
9.23.4.3	lines()	122
9.23.5	Member Data Documentation	122
9.23.5.1	column	123
9.23.5.2	filename	123
9.23.5.3	line	123
9.24	yy::position Class Reference	123
9.24.1	Detailed Description	124
9.24.2	Member Typedef Documentation	124
9.24.2.1	counter_type	124
9.24.2.2	filename_type	124
9.24.3	Constructor & Destructor Documentation	124
9.24.3.1	position()	124
9.24.4	Member Function Documentation	125
9.24.4.1	columns()	125
9.24.4.2	initialize()	125
9.24.4.3	lines()	125

9.24.5 Member Data Documentation	125
9.24.5.1 column	125
9.24.5.2 filename	125
9.24.5.3 line	126
9.25 productionnode Class Reference	126
9.25.1 Constructor & Destructor Documentation	127
9.25.1.1 productionnode() [1/2]	128
9.25.1.2 productionnode() [2/2]	128
9.25.1.3 ~productionnode()	128
9.25.2 Member Function Documentation	128
9.25.2.1 clone()	128
9.25.2.2 createRows()	128
9.25.2.3 dump()	128
9.25.2.4 fixSkips()	129
9.25.2.5 getName()	129
9.25.2.6 getSubsume()	129
9.25.2.7 optimize()	129
9.25.2.8 place()	129
9.25.2.9 subsume()	129
9.25.2.10 texName()	130
9.26 railnode Class Reference	130
9.26.1 Constructor & Destructor Documentation	131
9.26.1.1 railnode() [1/3]	132
9.26.1.2 railnode() [2/3]	132
9.26.1.3 railnode() [3/3]	132
9.26.1.4 ~railnode()	132
9.26.2 Member Function Documentation	132
9.26.2.1 clone()	132
9.26.2.2 dump()	132
9.26.2.3 getBottom()	133
9.26.2.4 getRailDir()	133
9.26.2.5 operator!=()	133
9.26.2.6 operator==()	133
9.26.2.7 place()	133
9.26.2.8 setBottom()	134
9.26.2.9 setRailDir()	134
9.26.2.10 texName()	134
9.26.3 Member Data Documentation	134
9.26.3.1 bottom	134
9.26.3.2 direction	134
9.26.3.3 side	134
9.26.3.4 top	135

9.27 rownode Class Reference	135
9.27.1 Constructor & Destructor Documentation	136
9.27.1.1 rownode() [1/2]	136
9.27.1.2 rownode() [2/2]	137
9.27.1.3 ~rownode()	137
9.27.2 Member Function Documentation	137
9.27.2.1 clone()	137
9.27.2.2 dump()	137
9.27.2.3 place()	137
9.27.2.4 texName()	138
9.28 annot::parser::semantic_type Class Reference	138
9.28.1 Detailed Description	139
9.28.2 Member Typedef Documentation	139
9.28.2.1 self_type	139
9.28.3 Constructor & Destructor Documentation	139
9.28.3.1 semantic_type() [1/2]	139
9.28.3.2 semantic_type() [2/2]	139
9.28.3.3 ~semantic_type()	139
9.28.4 Member Function Documentation	140
9.28.4.1 as() [1/2]	140
9.28.4.2 as() [2/2]	140
9.28.4.3 build() [1/2]	140
9.28.4.4 build() [2/2]	140
9.28.4.5 copy()	140
9.28.4.6 destroy()	141
9.28.4.7 emplace() [1/2]	141
9.28.4.8 emplace() [2/2]	141
9.28.4.9 move()	141
9.28.4.10 swap()	141
9.28.5 Member Data Documentation	142
9.28.5.1 yyalign_me	142
9.28.5.2 yyraw	142
9.29 singlenode Class Reference	142
9.29.1 Constructor & Destructor Documentation	144
9.29.1.1 singlenode() [1/2]	144
9.29.1.2 singlenode() [2/2]	144
9.29.1.3 ~singlenode()	144
9.29.2 Member Function Documentation	144
9.29.2.1 analyzeNonOptLoops()	144
9.29.2.2 analyzeOptLoops()	144
9.29.2.3 clone()	145
9.29.2.4 drawToLeftRail()	145

9.29.2.5 drawToRightRail()	145
9.29.2.6 fixSkips()	145
9.29.2.7 forgetChild()	145
9.29.2.8 getChild()	146
9.29.2.9 liftConcats()	146
9.29.2.10 mergeChoices()	146
9.29.2.11 mergeConcats()	146
9.29.2.12 mergeRails()	146
9.29.2.13 numChildren()	146
9.29.2.14 operator"!=(147
9.29.2.15 operator==(147
9.29.2.16 setNext()	147
9.29.2.17 setParent()	147
9.29.2.18 setPrevious()	147
9.29.2.19 subsume()	148
9.29.2.20 texName()	148
9.29.3 Member Data Documentation	148
9.29.3.1 body	148
9.30 annot::parser::stack< T, S >::slice Class Reference	148
9.30.1 Detailed Description	149
9.30.2 Constructor & Destructor Documentation	149
9.30.2.1 slice()	149
9.30.3 Member Function Documentation	149
9.30.3.1 operator[]()	149
9.31 yy::parser::stack< T, S >::slice Class Reference	149
9.31.1 Detailed Description	150
9.31.2 Constructor & Destructor Documentation	150
9.31.2.1 slice()	150
9.31.3 Member Function Documentation	150
9.31.3.1 operator[]()	150
9.32 annot::parser::symbol_kind Struct Reference	150
9.32.1 Detailed Description	151
9.32.2 Member Enumeration Documentation	151
9.32.2.1 symbol_kind_type	151
9.33 yy::parser::symbol_kind Struct Reference	151
9.33.1 Detailed Description	152
9.33.2 Member Enumeration Documentation	152
9.33.2.1 symbol_kind_type	152
9.34 annot::parser::symbol_type Struct Reference	153
9.34.1 Detailed Description	154
9.34.2 Member Typedef Documentation	154
9.34.2.1 super_type	154

9.34.3 Constructor & Destructor Documentation	154
9.34.3.1 symbol_type() [1/3]	154
9.34.3.2 symbol_type() [2/3]	155
9.34.3.3 symbol_type() [3/3]	155
9.35 yy::parser::symbol_type Struct Reference	155
9.35.1 Detailed Description	156
9.35.2 Member Typedef Documentation	156
9.35.2.1 super_type	157
9.35.3 Constructor & Destructor Documentation	157
9.35.3.1 symbol_type() [1/3]	157
9.35.3.2 symbol_type() [2/3]	157
9.35.3.3 symbol_type() [3/3]	157
9.36 annot::parser::syntax_error Struct Reference	158
9.36.1 Detailed Description	159
9.36.2 Constructor & Destructor Documentation	159
9.36.2.1 syntax_error() [1/2]	159
9.36.2.2 syntax_error() [2/2]	159
9.36.2.3 ~syntax_error()	159
9.36.3 Member Data Documentation	159
9.36.3.1 location	159
9.37 yy::parser::syntax_error Struct Reference	160
9.37.1 Detailed Description	161
9.37.2 Constructor & Destructor Documentation	161
9.37.2.1 syntax_error() [1/2]	161
9.37.2.2 syntax_error() [2/2]	161
9.37.2.3 ~syntax_error()	161
9.37.3 Member Data Documentation	161
9.37.3.1 location	161
9.38 termnode Class Reference	162
9.38.1 Constructor & Destructor Documentation	163
9.38.1.1 termnode() [1/2]	163
9.38.1.2 termnode() [2/2]	163
9.38.1.3 ~termnode()	163
9.38.2 Member Function Documentation	163
9.38.2.1 clone()	163
9.39 annot::parser::token Struct Reference	164
9.39.1 Detailed Description	164
9.39.2 Member Typedef Documentation	164
9.39.2.1 yytokentype	164
9.39.3 Member Enumeration Documentation	164
9.39.3.1 token_kind_type	164
9.40 yy::parser::token Struct Reference	165

9.40.1 Detailed Description	165
9.40.2 Member Typedef Documentation	165
9.40.2.1 yytokentype	165
9.40.3 Member Enumeration Documentation	165
9.40.3.1 token_kind_type	165
9.41 yy::parser::value_type Class Reference	166
9.41.1 Detailed Description	167
9.41.2 Member Typedef Documentation	167
9.41.2.1 self_type	167
9.41.3 Constructor & Destructor Documentation	167
9.41.3.1 value_type() [1/2]	168
9.41.3.2 value_type() [2/2]	168
9.41.3.3 ~value_type()	168
9.41.4 Member Function Documentation	168
9.41.4.1 as() [1/2]	168
9.41.4.2 as() [2/2]	168
9.41.4.3 build() [1/2]	169
9.41.4.4 build() [2/2]	169
9.41.4.5 copy()	169
9.41.4.6 destroy()	169
9.41.4.7 emplace() [1/2]	169
9.41.4.8 emplace() [2/2]	169
9.41.4.9 move()	170
9.41.4.10 swap()	170
9.41.5 Member Data Documentation	170
9.41.5.1 yyalign_me_	170
9.41.5.2 yyraw_	170
9.42 yy_buffer_state Struct Reference	170
9.42.1 Member Data Documentation	171
9.42.1.1 yy_at_bol	171
9.42.1.2 yy_bs_column	171
9.42.1.3 yy_bs_lineno	171
9.42.1.4 yy_buf_pos	171
9.42.1.5 yy_buf_size	171
9.42.1.6 yy_buffer_status	172
9.42.1.7 yy_ch_buf	172
9.42.1.8 yy_fill_buffer	172
9.42.1.9 yy_input_file	172
9.42.1.10 yy_is_interactive	172
9.42.1.11 yy_is_our_buffer	172
9.42.1.12 yy_n_chars	172
9.43 yy_trans_info Struct Reference	173

9.43.1 Member Data Documentation	173
9.43.1.1 yy_nxt	173
9.43.1.2 yy_verify	173
10 File Documentation	175
10.1 annot_lexer.cc File Reference	175
10.1.1 Macro Definition Documentation	179
10.1.1.1 annot_create_buffer_ALREADY_DEFINED	179
10.1.1.2 annot_delete_buffer_ALREADY_DEFINED	179
10.1.1.3 annot_flex_debug_ALREADY_DEFINED	180
10.1.1.4 annot_flush_buffer_ALREADY_DEFINED	180
10.1.1.5 annot_init_buffer_ALREADY_DEFINED	180
10.1.1.6 annot_load_buffer_state_ALREADY_DEFINED	180
10.1.1.7 annot_scan_buffer_ALREADY_DEFINED	180
10.1.1.8 annot_scan_bytes_ALREADY_DEFINED	180
10.1.1.9 annot_scan_string_ALREADY_DEFINED	180
10.1.1.10 annot_switch_to_buffer_ALREADY_DEFINED	180
10.1.1.11 annotalloc_ALREADY_DEFINED	181
10.1.1.12 annotensure_buffer_stack_ALREADY_DEFINED	181
10.1.1.13 annotfree_ALREADY_DEFINED	181
10.1.1.14 annotin_ALREADY_DEFINED	181
10.1.1.15 annotleng_ALREADY_DEFINED	181
10.1.1.16 annotlex_ALREADY_DEFINED	181
10.1.1.17 annotlineno_ALREADY_DEFINED	181
10.1.1.18 annotout_ALREADY_DEFINED	181
10.1.1.19 annotpop_buffer_state_ALREADY_DEFINED	182
10.1.1.20 annotpush_buffer_state_ALREADY_DEFINED	182
10.1.1.21 annotrealloc_ALREADY_DEFINED	182
10.1.1.22 annotrestart_ALREADY_DEFINED	182
10.1.1.23 annottext_ALREADY_DEFINED	182
10.1.1.24 annotwrap	182
10.1.1.25 annotwrap_ALREADY_DEFINED	182
10.1.1.26 BEGIN	182
10.1.1.27 ECHO	183
10.1.1.28 EOB_ACT_CONTINUE_SCAN	183
10.1.1.29 EOB_ACT_END_OF_FILE	183
10.1.1.30 EOB_ACT_LAST_MATCH	183
10.1.1.31 FLEX_BETA	183
10.1.1.32 FLEX_SCANNER	183
10.1.1.33 FLEXINT_H	183
10.1.1.34 INITIAL	183
10.1.1.35 INT16_MAX	184

10.1.1.36 INT16_MIN	184
10.1.1.37 INT32_MAX	184
10.1.1.38 INT32_MIN	184
10.1.1.39 INT8_MAX	184
10.1.1.40 INT8_MIN	184
10.1.1.41 REJECT	184
10.1.1.42 SIZE_MAX	184
10.1.1.43 UINT16_MAX	185
10.1.1.44 UINT32_MAX	185
10.1.1.45 UINT8_MAX	185
10.1.1.46 unput	185
10.1.1.47 YY_AT_BOL	185
10.1.1.48 YY_BREAK	185
10.1.1.49 YY_BUF_SIZE	185
10.1.1.50 YY_BUFFER_EOF_PENDING	186
10.1.1.51 YY_BUFFER_NEW	186
10.1.1.52 YY_BUFFER_NORMAL	186
10.1.1.53 yy_create_buffer	186
10.1.1.54 YY_CURRENT_BUFFER	186
10.1.1.55 YY_CURRENT_BUFFER_LVALUE	186
10.1.1.56 YY_DECL	186
10.1.1.57 yy_delete_buffer	187
10.1.1.58 YY_DO_BEFORE_ACTION	187
10.1.1.59 YY_END_OF_BUFFER	187
10.1.1.60 YY_END_OF_BUFFER_CHAR	187
10.1.1.61 YY_EXIT_FAILURE	187
10.1.1.62 YY_EXTRA_TYPE	187
10.1.1.63 YY_FATAL_ERROR	187
10.1.1.64 yy_flex_debug	188
10.1.1.65 YY_FLEX_MAJOR_VERSION	188
10.1.1.66 YY_FLEX_MINOR_VERSION	188
10.1.1.67 YY_FLEX_SUBMINOR_VERSION	188
10.1.1.68 yy_flush_buffer	188
10.1.1.69 YY_FLUSH_BUFFER	188
10.1.1.70 yy_init_buffer	188
10.1.1.71 YY_INPUT	189
10.1.1.72 YY_INT_ALIGNED	189
10.1.1.73 YY_LESS_LINENO	189
10.1.1.74 YY_LINENO_REWIND_TO	189
10.1.1.75 yy_load_buffer_state	190
10.1.1.76 YY_MORE_ADJ	190
10.1.1.77 yy_new_buffer	190

10.1.1.78 YY_NEW_FILE	190
10.1.1.79 YY_NO_INPUT	190
10.1.1.80 YY_NULL	190
10.1.1.81 YY_NUM_RULES	190
10.1.1.82 YY_READ_BUF_SIZE	191
10.1.1.83 YY_RESTORE_YY_MORE_OFFSET	191
10.1.1.84 YY_RULE_SETUP	191
10.1.1.85 YY_SC_TO_UI	191
10.1.1.86 yy_scan_buffer	191
10.1.1.87 yy_scan_bytes	191
10.1.1.88 yy_scan_string	191
10.1.1.89 yy_set_bol	192
10.1.1.90 yy_set_interactive	192
10.1.1.91 YY_SKIP_YYWRAP	192
10.1.1.92 YY_START	192
10.1.1.93 YY_START_STACK_INCR	192
10.1.1.94 YY_STATE_BUF_SIZE	193
10.1.1.95 YY_STATE_EOF	193
10.1.1.96 YY_STRUCT_YY_BUFFER_STATE	193
10.1.1.97 yy_switch_to_buffer	193
10.1.1.98 YY_TYPEDEF_YY_BUFFER_STATE	193
10.1.1.99 YY_TYPEDEF_YY_SIZE_T	193
10.1.1.100 YY_USER_ACTION	193
10.1.1.101 yyalloc	194
10.1.1.102 yyconst	194
10.1.1.103 yyensure_buffer_stack	194
10.1.1.104 yyfree	194
10.1.1.105 yyget_debug	194
10.1.1.106 yyget_extra	194
10.1.1.107 yyget_in	194
10.1.1.108 yyget_leng	195
10.1.1.109 yyget_lineno	195
10.1.1.110 yyget_out	195
10.1.1.111 yyget_text	195
10.1.1.112 yyin	195
10.1.1.113 yyleng	195
10.1.1.114 yyless [1/2]	196
10.1.1.115 yyless [2/2]	196
10.1.1.116 yylex	196
10.1.1.117 yylex_destroy	196
10.1.1.118 yylex_init	196
10.1.1.119 yylex_init_extra	197

10.1.1.120	yylineno	197
10.1.1.121	yymore	197
10.1.1.122	yynoreturn	197
10.1.1.123	yyout	197
10.1.1.124	yypop_buffer_state	197
10.1.1.125	yypush_buffer_state	197
10.1.1.126	yyrealloc	198
10.1.1.127	yyrestart	198
10.1.1.128	yyset_debug	198
10.1.1.129	yyset_extra	198
10.1.1.130	yyset_in	198
10.1.1.131	yyset_lineno	198
10.1.1.132	yyset_out	198
10.1.1.133	YYSTATE	198
10.1.1.134	YYTABLES_NAME	199
10.1.1.135	yyterminate	199
10.1.1.136	yytext	199
10.1.1.137	yytext_ptr	199
10.1.1.138	yywrap	199
10.1.2	Typedef Documentation	199
10.1.2.1	flex_int16_t	199
10.1.2.2	flex_int32_t	199
10.1.2.3	flex_int8_t	200
10.1.2.4	flex_uint16_t	200
10.1.2.5	flex_uint32_t	200
10.1.2.6	flex_uint8_t	200
10.1.2.7	YY_BUFFER_STATE	200
10.1.2.8	YY_CHAR	200
10.1.2.9	yy_size_t	200
10.1.2.10	yy_state_type	200
10.1.3	Function Documentation	201
10.1.3.1	if()	201
10.1.3.2	step()	201
10.1.3.3	stripquotes()	201
10.1.3.4	while()	201
10.1.3.5	yy_create_buffer()	201
10.1.3.6	yy_delete_buffer()	202
10.1.3.7	yy_flush_buffer()	202
10.1.3.8	yy_scan_buffer()	202
10.1.3.9	yy_scan_bytes()	203
10.1.3.10	yy_scan_string()	203
10.1.3.11	yy_switch_to_buffer()	203

10.1.3.12 yyalloc()	204
10.1.3.13 yyfree()	204
10.1.3.14 yypush_buffer_state()	204
10.1.3.15 yyrealloc()	204
10.1.3.16 yyrestart()	204
10.1.3.17 yyset_debug()	205
10.1.3.18 yyset_extra()	205
10.1.3.19 yyset_in()	205
10.1.3.20 yyset_lineno()	205
10.1.3.21 yyset_out()	206
10.1.4 Variable Documentation	206
10.1.4.1 yy_act	206
10.1.4.2 yy_bp	206
10.1.4.3 yy_cp	206
10.1.4.4 YY_DECL	206
10.1.4.5 yy_flex_debug	207
10.1.4.6 yyin	207
10.1.4.7 yyleng	207
10.1.4.8 yylineno	207
10.1.4.9 yyout	207
10.1.4.10 yytext	207
10.2 annot_lexer.hh File Reference	208
10.2.1 Macro Definition Documentation	210
10.2.1.1 annotIN_HEADER	211
10.2.1.2 annotwrap	211
10.2.1.3 FLEX_BETA	211
10.2.1.4 FLEX_SCANNER	211
10.2.1.5 FLEXINT_H	211
10.2.1.6 INT16_MAX	211
10.2.1.7 INT16_MIN	211
10.2.1.8 INT32_MAX	211
10.2.1.9 INT32_MIN	212
10.2.1.10 INT8_MAX	212
10.2.1.11 INT8_MIN	212
10.2.1.12 SIZE_MAX	212
10.2.1.13 UINT16_MAX	212
10.2.1.14 UINT32_MAX	212
10.2.1.15 UINT8_MAX	212
10.2.1.16 YY_BUF_SIZE	212
10.2.1.17 yy_create_buffer	213
10.2.1.18 YY_DECL	213
10.2.1.19 YY_DECL_IS_OURS	213

10.2.1.20 <code>yy_delete_buffer</code>	213
10.2.1.21 <code>YY_EXTRA_TYPE</code>	213
10.2.1.22 <code>yy_flex_debug</code>	213
10.2.1.23 <code>YY_FLEX_MAJOR_VERSION</code>	213
10.2.1.24 <code>YY_FLEX_MINOR_VERSION</code>	213
10.2.1.25 <code>YY_FLEX_SUBMINOR_VERSION</code>	214
10.2.1.26 <code>yy_flush_buffer</code>	214
10.2.1.27 <code>yy_init_buffer</code>	214
10.2.1.28 <code>YY_INT_ALIGNED</code>	214
10.2.1.29 <code>yy_load_buffer_state</code>	214
10.2.1.30 <code>YY_READ_BUF_SIZE</code>	214
10.2.1.31 <code>yy_scan_buffer</code>	214
10.2.1.32 <code>yy_scan_bytes</code>	215
10.2.1.33 <code>yy_scan_string</code>	215
10.2.1.34 <code>YY_SKIP_YWRAP</code>	215
10.2.1.35 <code>YY_START_STACK_INCR</code>	215
10.2.1.36 <code>YY_STRUCT_YY_BUFFER_STATE</code>	215
10.2.1.37 <code>yy_switch_to_buffer</code>	215
10.2.1.38 <code>YY_TYPEDEF_YY_BUFFER_STATE</code>	215
10.2.1.39 <code>YY_TYPEDEF_YY_SIZE_T</code>	215
10.2.1.40 <code>yyalloc</code>	216
10.2.1.41 <code>yyconst</code>	216
10.2.1.42 <code>yyensure_buffer_stack</code>	216
10.2.1.43 <code>yyfree</code>	216
10.2.1.44 <code>yyget_debug</code>	216
10.2.1.45 <code>yyget_extra</code>	216
10.2.1.46 <code>yyget_in</code>	216
10.2.1.47 <code>yyget_leng</code>	217
10.2.1.48 <code>yyget_lineno</code>	217
10.2.1.49 <code>yyget_out</code>	217
10.2.1.50 <code>yyget_text</code>	217
10.2.1.51 <code>yyin</code>	217
10.2.1.52 <code>yylen</code>	217
10.2.1.53 <code>yylex</code>	217
10.2.1.54 <code>yylex_destroy</code>	218
10.2.1.55 <code>yylex_init</code>	218
10.2.1.56 <code>yylex_init_extra</code>	218
10.2.1.57 <code>yylineno</code>	218
10.2.1.58 <code>yynoreturn</code>	218
10.2.1.59 <code>yyout</code>	218
10.2.1.60 <code>yypop_buffer_state</code>	218
10.2.1.61 <code>yypush_buffer_state</code>	219

10.2.1.62	yyrealloc	219
10.2.1.63	yyrestart	219
10.2.1.64	yyset_debug	219
10.2.1.65	yyset_extra	219
10.2.1.66	yyset_in	219
10.2.1.67	yyset_lineno	219
10.2.1.68	yyset_out	219
10.2.1.69	yytext	220
10.2.1.70	yytext_ptr	220
10.2.1.71	yywrap	220
10.2.2	Typedef Documentation	220
10.2.2.1	flex_int16_t	220
10.2.2.2	flex_int32_t	220
10.2.2.3	flex_int8_t	220
10.2.2.4	flex_uint16_t	220
10.2.2.5	flex_uint32_t	221
10.2.2.6	flex_uint8_t	221
10.2.2.7	YY_BUFFER_STATE	221
10.2.2.8	yy_size_t	221
10.2.3	Function Documentation	221
10.2.3.1	yy_create_buffer()	221
10.2.3.2	yy_delete_buffer()	222
10.2.3.3	yy_flush_buffer()	222
10.2.3.4	yy_scan_buffer()	222
10.2.3.5	yy_scan_bytes()	222
10.2.3.6	yy_scan_string()	223
10.2.3.7	yy_switch_to_buffer()	223
10.2.3.8	yyalloc()	224
10.2.3.9	yyfree()	224
10.2.3.10	yypush_buffer_state()	224
10.2.3.11	yyrealloc()	224
10.2.3.12	yyrestart()	224
10.2.3.13	yyset_debug()	225
10.2.3.14	yyset_extra()	225
10.2.3.15	yyset_in()	225
10.2.3.16	yyset_lineno()	225
10.2.3.17	yyset_out()	226
10.2.4	Variable Documentation	226
10.2.4.1	yyin	226
10.2.4.2	yyleng	226
10.2.4.3	yylineno	226
10.2.4.4	yyout	226

10.2.4.5 ytext	226
10.3 annot_location.hh File Reference	227
10.3.1 Detailed Description	228
10.3.2 Macro Definition Documentation	228
10.3.2.1 YY_NULLPTR	228
10.4 annot_parser.cc File Reference	229
10.4.1 Macro Definition Documentation	230
10.4.1.1 YY_	230
10.4.1.2 YY_DECL	230
10.4.1.3 YY_EXCEPTIONS	230
10.4.1.4 YY_REDUCE_PRINT	230
10.4.1.5 YY_STACK_PRINT	230
10.4.1.6 YY_SYMBOL_PRINT	230
10.4.1.7 YYABORT	231
10.4.1.8 YYACCEPT	231
10.4.1.9 YYCASE_	231
10.4.1.10 YYCDEBUG	231
10.4.1.11 yyclearin	231
10.4.1.12 yyerrok	231
10.4.1.13 YYERROR	231
10.4.1.14 yylex	232
10.4.1.15 YYLLOC_DEFAULT	232
10.4.1.16 YYRECOVERING	232
10.4.1.17 YYRHSLOC	232
10.4.2 Function Documentation	232
10.4.2.1 scanAnnot()	232
10.4.3 Variable Documentation	233
10.4.3.1 aloc	233
10.4.3.2 loc	233
10.4.3.3 YY_DECL	233
10.5 annot_parser.hh File Reference	233
10.5.1 Detailed Description	235
10.5.2 Macro Definition Documentation	235
10.5.2.1 ANNOT_ASSERT	235
10.5.2.2 ANNOTDEBUG	235
10.5.2.3 YY_ATTRIBUTE_PURE	235
10.5.2.4 YY_ATTRIBUTE_UNUSED	235
10.5.2.5 YY_CAST	236
10.5.2.6 YY_CONSTEXPR	236
10.5.2.7 YY_COPY	236
10.5.2.8 YY_CPLUSPLUS	236
10.5.2.9 YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN	236

10.5.2.10 YY_IGNORE_MAYBE_UNINITIALIZED_END	236
10.5.2.11 YY_IGNORE_USELESS_CAST_BEGIN	236
10.5.2.12 YY_IGNORE_USELESS_CAST_END	237
10.5.2.13 YY_INITIAL_VALUE	237
10.5.2.14 YY_MOVE	237
10.5.2.15 YY_MOVE_OR_COPY	237
10.5.2.16 YY_MOVE_REF	237
10.5.2.17 YY_NOEXCEPT	237
10.5.2.18 YY_NOTHROW	237
10.5.2.19 YY_REINTERPRET_CAST	238
10.5.2.20 YY_RVREF	238
10.5.2.21 YY_USE	238
10.6 CODE_OF_CONDUCT.md File Reference	238
10.7 CONTRIBUTING.md File Reference	238
10.8 driver.cc File Reference	238
10.9 driver.hh File Reference	239
10.9.1 Macro Definition Documentation	240
10.9.1.1 YY_DECL	240
10.9.2 Variable Documentation	240
10.9.2.1 YY_DECL	240
10.10 graph.cc File Reference	240
10.10.1 Function Documentation	241
10.10.1.1 latexwrite()	241
10.10.1.2 nextChain()	241
10.10.1.3 nextCoord()	241
10.10.1.4 nextFit()	241
10.10.1.5 nextNode()	241
10.10.1.6 stripSpecial()	242
10.11 graph.hh File Reference	242
10.11.1 Typedef Documentation	243
10.11.1.1 annot_t	243
10.11.1.2 annotmap	243
10.11.2 Enumeration Type Documentation	243
10.11.2.1 vraildir	243
10.11.2.2 vrailside	244
10.11.3 Function Documentation	244
10.11.3.1 latexwrite()	244
10.11.3.2 nextChain()	244
10.11.3.3 nextCoord()	244
10.11.3.4 nextFit()	244
10.11.3.5 nextNode()	245
10.11.3.6 stripSpecial()	245

10.12 lexer.cc File Reference	245
10.12.1 Macro Definition Documentation	248
10.12.1.1 A	248
10.12.1.2 BEGIN	248
10.12.1.3 ECHO	249
10.12.1.4 EOB_ACT_CONTINUE_SCAN	249
10.12.1.5 EOB_ACT_END_OF_FILE	249
10.12.1.6 EOB_ACT_LAST_MATCH	249
10.12.1.7 FLEX_BETA	249
10.12.1.8 FLEX_DEBUG	249
10.12.1.9 FLEX_SCANNER	249
10.12.1.10 FLEXINT_H	249
10.12.1.11 INITIAL	250
10.12.1.12 INT16_MAX	250
10.12.1.13 INT16_MIN	250
10.12.1.14 INT32_MAX	250
10.12.1.15 INT32_MIN	250
10.12.1.16 INT8_MAX	250
10.12.1.17 INT8_MIN	250
10.12.1.18 REJECT	250
10.12.1.19 SIZE_MAX	251
10.12.1.20 UINT16_MAX	251
10.12.1.21 UINT32_MAX	251
10.12.1.22 UINT8_MAX	251
10.12.1.23 unput	251
10.12.1.24 YY_AT_BOL	251
10.12.1.25 YY_BREAK	251
10.12.1.26 YY_BUF_SIZE	252
10.12.1.27 YY_BUFFER_EOF_PENDING	252
10.12.1.28 YY_BUFFER_NEW	252
10.12.1.29 YY_BUFFER_NORMAL	252
10.12.1.30 YY_CURRENT_BUFFER	252
10.12.1.31 YY_CURRENT_BUFFER_LVALUE	252
10.12.1.32 YY_DO_BEFORE_ACTION	252
10.12.1.33 YY_END_OF_BUFFER	253
10.12.1.34 YY_END_OF_BUFFER_CHAR	253
10.12.1.35 YY_EXIT_FAILURE	253
10.12.1.36 YY_EXTRA_TYPE	253
10.12.1.37 YY_FATAL_ERROR	253
10.12.1.38 YY_FLEX_MAJOR_VERSION	253
10.12.1.39 YY_FLEX_MINOR_VERSION	253
10.12.1.40 YY_FLEX_SUBMINOR_VERSION	254

10.12.1.41 YY_FLUSH_BUFFER	254
10.12.1.42 YY_INPUT	254
10.12.1.43 YY_INT_ALIGNED	254
10.12.1.44 YY_LESS_LINENO	254
10.12.1.45 YY_LINENO_REWIND_TO	254
10.12.1.46 YY_MORE_ADJ	254
10.12.1.47 yy_new_buffer	255
10.12.1.48 YY_NEW_FILE	255
10.12.1.49 YY_NO_INPUT	255
10.12.1.50 YY_NULL	255
10.12.1.51 YY_NUM_RULES	255
10.12.1.52 YY_READ_BUF_SIZE	255
10.12.1.53 YY_RESTORE YY_MORE_OFFSET	255
10.12.1.54 YY_RULE_SETUP	255
10.12.1.55 YY_SC_TO_UI	256
10.12.1.56 yy_set_bol	256
10.12.1.57 yy_set_interactive	256
10.12.1.58 YY_SKIP_YYWRAP	256
10.12.1.59 YY_START	256
10.12.1.60 YY_START_STACK_INCR	257
10.12.1.61 YY_STATE_BUF_SIZE	257
10.12.1.62 YY_STATE_EOF	257
10.12.1.63 YY_STRUCT_YY_BUFFER_STATE	257
10.12.1.64 YY_TYPEDEF_YY_BUFFER_STATE	257
10.12.1.65 YY_TYPEDEF_YY_SIZE_T	257
10.12.1.66 YY_USER_ACTION	257
10.12.1.67 yyconst	258
10.12.1.68 yyless [1/2]	258
10.12.1.69 yyless [2/2]	258
10.12.1.70 yymore	258
10.12.1.71 yynoreturn	258
10.12.1.72 YYSTATE	259
10.12.1.73 YYTABLES_NAME	259
10.12.1.74 yyterminate	259
10.12.1.75 yytext_ptr	259
10.12.1.76 yywrap	259
10.12.2 Typedef Documentation	259
10.12.2.1 flex_int16_t	259
10.12.2.2 flex_int32_t	259
10.12.2.3 flex_int8_t	260
10.12.2.4 flex_uint16_t	260
10.12.2.5 flex_uint32_t	260

10.12.2.6 flex_uint8_t	260
10.12.2.7 YY_BUFFER_STATE	260
10.12.2.8 YY_CHAR	260
10.12.2.9 yy_size_t	260
10.12.2.10 yy_state_type	260
10.12.3 Function Documentation	261
10.12.3.1 if()	261
10.12.3.2 step()	261
10.12.3.3 while()	261
10.12.3.4 yy_create_buffer()	261
10.12.3.5 yy_delete_buffer()	261
10.12.3.6 yy_flush_buffer()	262
10.12.3.7 yy_scan_buffer()	262
10.12.3.8 yy_scan_bytes()	262
10.12.3.9 yy_scan_string()	263
10.12.3.10 yy_switch_to_buffer()	263
10.12.3.11 yyalloc()	264
10.12.3.12 yyfree()	264
10.12.3.13 yyget_debug()	264
10.12.3.14 yyget_extra()	264
10.12.3.15 yyget_in()	264
10.12.3.16 yyget_leng()	264
10.12.3.17 yyget_lineno()	264
10.12.3.18 yyget_out()	265
10.12.3.19 yyget_text()	265
10.12.3.20 yylex_destroy()	265
10.12.3.21 yypop_buffer_state()	265
10.12.3.22 yypush_buffer_state()	265
10.12.3.23 yyrealloc()	265
10.12.3.24 yyrestart()	266
10.12.3.25 yyset_debug()	266
10.12.3.26 yyset_extra()	266
10.12.3.27 yyset_in()	266
10.12.3.28 yyset_lineno()	267
10.12.3.29 yyset_out()	267
10.12.4 Variable Documentation	267
10.12.4.1 subloc	267
10.12.4.2 yy_act	267
10.12.4.3 yy_bp	267
10.12.4.4 yy_cp	268
10.12.4.5 YY_DECL	268
10.12.4.6 yy_flex_debug	268

10.12.4.7 yyin	268
10.12.4.8 yyleng	268
10.12.4.9 yylineno	268
10.12.4.10 yyout	268
10.12.4.11 yytext	269
10.13 lexer.hh File Reference	269
10.13.1 Macro Definition Documentation	271
10.13.1.1 FLEX_BETA	271
10.13.1.2 FLEX_DEBUG	272
10.13.1.3 FLEX_SCANNER	272
10.13.1.4 FLEXINT_H	272
10.13.1.5 INT16_MAX	272
10.13.1.6 INT16_MIN	272
10.13.1.7 INT32_MAX	272
10.13.1.8 INT32_MIN	272
10.13.1.9 INT8_MAX	272
10.13.1.10 INT8_MIN	273
10.13.1.11 SIZE_MAX	273
10.13.1.12 UINT16_MAX	273
10.13.1.13 UINT32_MAX	273
10.13.1.14 UINT8_MAX	273
10.13.1.15 YY_BUF_SIZE	273
10.13.1.16 YY_DECL	273
10.13.1.17 YY_DECL_IS_OURS	273
10.13.1.18 YY_EXTRA_TYPE	274
10.13.1.19 YY_FLEX_MAJOR_VERSION	274
10.13.1.20 YY_FLEX_MINOR_VERSION	274
10.13.1.21 YY_FLEX_SUBMINOR_VERSION	274
10.13.1.22 YY_INT_ALIGNED	274
10.13.1.23 YY_READ_BUF_SIZE	274
10.13.1.24 YY_SKIP_YYWRAP	274
10.13.1.25 YY_START_STACK_INCR	274
10.13.1.26 YY_STRUCT_YY_BUFFER_STATE	275
10.13.1.27 YY_TYPEDEF_YY_BUFFER_STATE	275
10.13.1.28 YY_TYPEDEF_YY_SIZE_T	275
10.13.1.29 yyconst	275
10.13.1.30 yyIN_HEADER	275
10.13.1.31 yynoreturn	275
10.13.1.32 yytext_ptr	275
10.13.1.33 yywrap	275
10.13.2 Typedef Documentation	276
10.13.2.1 flex_int16_t	276

10.13.2.2 flex_int32_t	276
10.13.2.3 flex_int8_t	276
10.13.2.4 flex_uint16_t	276
10.13.2.5 flex_uint32_t	276
10.13.2.6 flex_uint8_t	276
10.13.2.7 YY_BUFFER_STATE	276
10.13.2.8 yy_size_t	277
10.13.3 Function Documentation	277
10.13.3.1 yy_create_buffer()	277
10.13.3.2 yy_delete_buffer()	277
10.13.3.3 yy_flush_buffer()	277
10.13.3.4 yy_scan_buffer()	278
10.13.3.5 yy_scan_bytes()	278
10.13.3.6 yy_scan_string()	278
10.13.3.7 yy_switch_to_buffer()	279
10.13.3.8 yyallocc()	279
10.13.3.9 yyfree()	279
10.13.3.10 yyget_debug()	279
10.13.3.11 yyget_extra()	280
10.13.3.12 yyget_in()	280
10.13.3.13 yyget_leng()	280
10.13.3.14 yyget_lineno()	280
10.13.3.15 yyget_out()	280
10.13.3.16 yyget_text()	280
10.13.3.17 yylex()	280
10.13.3.18 yylex_destroy()	281
10.13.3.19 yypop_buffer_state()	281
10.13.3.20 yypush_buffer_state()	281
10.13.3.21 yyrealloc()	281
10.13.3.22 yyrestart()	281
10.13.3.23 yyset_debug()	282
10.13.3.24 yyset_extra()	282
10.13.3.25 yyset_in()	282
10.13.3.26 yyset_lineno()	282
10.13.3.27 yyset_out()	283
10.13.4 Variable Documentation	283
10.13.4.1 yyin	283
10.13.4.2 yyleng	283
10.13.4.3 yylineno	283
10.13.4.4 yyout	283
10.13.4.5 yytext	283
10.14 location.hh File Reference	284

10.14.1 Detailed Description	285
10.14.2 Macro Definition Documentation	285
10.14.2.1 YY_NULLPTR	286
10.15 main.cc File Reference	286
10.15.1 Function Documentation	286
10.15.1.1 main()	286
10.15.1.2 usage()	287
10.15.2 Variable Documentation	287
10.15.2.1 description	287
10.15.2.2 options	287
10.15.2.3 optstring	287
10.16 nodesize.hh File Reference	288
10.17 optimize.cc File Reference	288
10.18 output.cc File Reference	289
10.19 parser.cc File Reference	290
10.19.1 Macro Definition Documentation	291
10.19.1.1 YY_	291
10.19.1.2 YY_EXCEPTIONS	291
10.19.1.3 YY_REDUCE_PRINT	291
10.19.1.4 YY_STACK_PRINT	291
10.19.1.5 YY_SYMBOL_PRINT	291
10.19.1.6 YYABORT	292
10.19.1.7 YYACCEPT	292
10.19.1.8 YYCASE_	292
10.19.1.9 YYCDEBUG	292
10.19.1.10 yyclearin	292
10.19.1.11 yyerrok	292
10.19.1.12 YYERROR	292
10.19.1.13 YYLLOC_DEFAULT	293
10.19.1.14 YYRECOVERING	293
10.19.1.15 YYRHSLOC	293
10.19.2 Function Documentation	293
10.19.2.1 wrapChoice()	293
10.19.3 Variable Documentation	293
10.19.3.1 loc	293
10.20 parser.hh File Reference	294
10.20.1 Detailed Description	295
10.20.2 Macro Definition Documentation	296
10.20.2.1 YY_ASSERT	296
10.20.2.2 YY_ATTRIBUTE_PURE	296
10.20.2.3 YY_ATTRIBUTE_UNUSED	296
10.20.2.4 YY_CAST	296

10.20.2.5 YY_CONSTEXPR	296
10.20.2.6 YY_COPY	296
10.20.2.7 YY_CPLUSPLUS	297
10.20.2.8 YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN	297
10.20.2.9 YY_IGNORE_MAYBE_UNINITIALIZED_END	297
10.20.2.10 YY_IGNORE_USELESS_CAST_BEGIN	297
10.20.2.11 YY_IGNORE_USELESS_CAST_END	297
10.20.2.12 YY_INITIAL_VALUE	297
10.20.2.13 YY_MOVE	297
10.20.2.14 YY_MOVE_OR_COPY	298
10.20.2.15 YY_MOVE_REF	298
10.20.2.16 YY_NOEXCEPT	298
10.20.2.17 YY_NOTHROW	298
10.20.2.18 YY_REINTERPRET_CAST	298
10.20.2.19 YY_RVREF	298
10.20.2.20 YY_USE	298
10.20.2.21 YYDEBUG	299
10.20.3 Function Documentation	299
10.20.3.1 scanAnnot()	299
10.21 README.md File Reference	299
10.22 subsume.cc File Reference	299
10.22.1 Macro Definition Documentation	300
10.22.1.1 ARRAY_SIZE	300
10.23 util.cc File Reference	300
10.23.1 Function Documentation	300
10.23.1.1 camelcase()	300
10.24 util.hh File Reference	301
10.24.1 Function Documentation	301
10.24.1.1 camelcase()	301

Chapter 1

Contributor Covenant Code of Conduct

1.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

1.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

1.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

1.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

1.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at . All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

1.6 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](https://www.contributor-covenant.org/version/1/4/code-of-conduct.html), version 1.4, available at <https://www.contributor-covenant.org/version/1/4/code-of-conduct.html>

For answers to common questions about this code of conduct, see <https://www.contributor-covenant.org/faq>

Chapter 2

CONTRIBUTING

To contribute, create a pull request, then email me at larry.pyeatt@sdsmt.edu

Chapter 3

ebnf2tikz

Author: Larry D. Pyeatt

February, 2021

3.1 What Does It Do?

It is an optimizing compiler that converts (possibly annotated) **Extended Backus-Naur Form** (EBNF) to railroad diagrams expressed as LaTeX **TikZ** commands.

For example, if you feed a file containing the following annotated EBNF into ebnf2tikz:

```
case_statement_alternative =
  'when' , choices , '=>', sequence_of_statements;
subsume
choices =
  choice , { '|' , choice } ;
subsume
choice =
  simple_expression |
  discrete_range |
  simple_name |
  'others' ;
```

Then it will output the following TikZ code:

```
\begin{figure}
\centerline{
\begin{tikzpicture}
\node at (0pt,0pt)[anchor=west](name){\railname{case\_statement\_alternative\strut}};
\coordinate (node42) at (59.3677pt,-21pt);
\coordinate (node42linetop) at (59.3677pt,-27pt);
\coordinate (node42linebottom) at (59.3677pt,-101pt);
\draw [rounded corners=\railcorners] (node42linetop) -- (node42linebottom);
\draw [rounded corners=\railcorners] (node42linetop) -- (node42) -- +(east:8pt);
\coordinate (node45) at (67.3677pt,-21pt);
\coordinate (node45linetop) at (67.3677pt,-27pt);
\coordinate (node45linebottom) at (67.3677pt,-79pt);
\draw [rounded corners=\railcorners] (node45linetop) -- (node45linebottom);
\draw [rounded corners=\railcorners] (node45linetop) -- (node45) -- +(west:8pt);
\coordinate (node51) at (156.118pt,-21pt);
\coordinate (node51linetop) at (156.118pt,-27pt);
\coordinate (node51linebottom) at (156.118pt,-79pt);
\draw [rounded corners=\railcorners] (node51linetop) -- (node51linebottom);
\draw [rounded corners=\railcorners] (node51linetop) -- (node51) -- +(east:8pt);
\coordinate (node53) at (164.118pt,-21pt);
\coordinate (node53linetop) at (164.118pt,-27pt);
\coordinate (node53linebottom) at (164.118pt,-101pt);
\draw [rounded corners=\railcorners] (node53linetop) -- (node53linebottom);
\draw [rounded corners=\railcorners] (node53linetop) -- (node53) -- +(west:8pt);
\node (node1) at (16pt,-21pt)[anchor=west,terminal] {\railtermname{when\strut}};
\writenodesize{node1}
\draw [rounded corners=\railcorners] (node1.east) -- (node42.west);
\node (node47) at (75.3677pt,-21pt)[anchor=west,nonterminal] {\railname{simple\_expression\strut}};
```

```

\writenodesize{node47}
\node (node48) at (83.5977pt,-43pt) [anchor=west,nonterminal] {\railname{discrete\_range\strut}};
\writenodesize{node48}
\node (node49) at (84.5176pt,-65pt) [anchor=west,nonterminal] {\railname{simple\_name\strut}};
\writenodesize{node49}
\node (node50) at (92.8288pt,-87pt) [anchor=west,terminal] {\railtermname{others\strut}};
\writenodesize{node50}
\draw [rounded corners=\railcorners] (node45.east) -- (node47.west);
\draw [rounded corners=\railcorners] (node47.east) -- (node51.west);
\node (node52) at (104.743pt,-109pt) [anchor=west,terminal] {\railtermname{||}};
\writenodesize{node52}
\draw [rounded corners=\railcorners] (node42.east) -- (node45.west);
\draw [rounded corners=\railcorners] (node51.east) -- (node53.west);
\node (node4) at (172.118pt,-21pt) [anchor=west,terminal] {\railtermname{=>\strut}};
\writenodesize{node4}
\draw [rounded corners=\railcorners] (node53.east) -- (node4.west);
\node (node5) at (199.775pt,-21pt) [anchor=west,nonterminal] {\railname{sequence\_of\_statements\strut}};
\writenodesize{node5}
\draw [rounded corners=\railcorners] (node4.east) -- (node5.west);
\draw [rounded corners=\railcorners] (node48.west) -- (node48.west-|node45) -- (node45linetop);
\draw [rounded corners=\railcorners] (node49.west) -- (node49.west-|node45) -- (node45linetop);
\draw [rounded corners=\railcorners] (node50.west) -- (node50.west-|node45) -- (node45linetop);
\draw [rounded corners=\railcorners] (node52.west) -- (node52.west-|node42) -- (node42linetop);
\draw [rounded corners=\railcorners] (node48.east) -- (node48.east-|node51) -- (node51linetop);
\draw [rounded corners=\railcorners] (node49.east) -- (node49.east-|node51) -- (node51linetop);
\draw [rounded corners=\railcorners] (node50.east) -- (node50.east-|node51) -- (node51linetop);
\draw [rounded corners=\railcorners] (node52.east) -- (node52.east-|node53) -- (node53linetop);
\end{tikzpicture}
}
\caption{No Caption.}
\label{No Caption.}
\end{figure}

```

You will need a `\usepackage{ebnf2tikz}` command in the preamble of your LaTeX document. Then you can just include the TikZ code in your LaTeX document, and it will draw this:

3.2 About the Code

This is a work in progress. There are still a couple of bugs that I am aware of, but nothing major. Some of the line drawing is not quite right, especially involving choice or loop nodes before and after newlines, and choices inside loops. I should have that fixed in a few days.

Originally, I planned to have TikZ do most of the work. However, while I could get it to do small diagrams, it failed miserably when the level of nesting went beyond three. TikZ really does not have the concept of "sub-images" that have user-defined anchor points. I think it is possible because, ... look at CircuiTikZ.

Trying to get TikZ to do all of the work to lay out complex diagrams was a nightmare. It does not do recursive structures well. I put in some effort, then gave up and decided to go another direction. I now have ebnf2tikz do all of the layout, and just use TikZ to do the drawing. This does mean that ebnf2tikz needs some information from LaTeX about how big the basic nodes are. Therefore, you have to run ebnf2tikz, then LaTeX, then ebnf2tikz again, then LaTeX again.

I have written it so that you can:

1. Run ebnf2tikz to produce all of the diagrams, but they are not correct.
2. Run the incorrect diagrams through LaTeX to get the dimensions of the basic nodes and the settings for railcolsep and railrowsep.
3. After that, re-run ebnf2tikz and all of the diagrams are correct.
4. Run LaTeX again and everything looks good.

Any change to the ebnf file requires these four steps to get everything looking good again. It is not so different from bibtex, makeindex, etc. The bottom line is that you may have to run ebnf2tikz twice if you change its input files, and you will have to run it at least once if you change the input file or change railrowsep or railcolsep or any other layout settings.

The good news is that this approach makes the diagrams as concise as they can possibly be. All of the layout is handled by ebnf2tikz, so LaTeX does not spend a lot of time on them. Also, I may be able to do some sort of "auto-newline" thing.

All nodes are placed using exact coordinates `\node (nodename) at (exact coordinate)` but all lines are drawn using the node names.

3.3 To Do

I have not written the ebnf2tikz style file, so the ``

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

annot	17
yy	20

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

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

Base	
annot::parser::basic_symbol< Base >	25
yy::parser::basic_symbol< Base >	30
yy::parser::symbol_type	155
annot::parser::basic_symbol< by_state >	25
annot::parser::by_kind	36
annot::parser::basic_symbol< by_kind >	25
annot::parser::symbol_type	153
yy::parser::by_kind	38
annot::parser::context	50
yy::parser::context	51
coordinate	52
driver	54
grammar	56
annot::location	59
yy::location	62
node	80
multinode	70
choicenode	41
concatnode	45
loopnode	66
nontermnode	97
nullnode	103
termnode	162
railnode	130
newlinenode	77
singlenode	142
productionnode	126
rownode	135
nodesizes	95
annot::parser	105
yy::parser	112
annot::position	120
yy::position	123

std::runtime_error	
annot::parser::syntax_error	158
yy::parser::syntax_error	160
annot::parser::semantic_type	138
annot::parser::stack< T, S >::slice	148
yy::parser::stack< T, S >::slice	149
annot::parser::symbol_kind	150
yy::parser::symbol_kind	151
annot::parser::token	164
yy::parser::token	165
yy::parser::value_type	166
yy_buffer_state	170
yy_trans_info	173

Chapter 6

Class Index

6.1 Class List

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

annot::parser::basic_symbol< Base >	25
yy::parser::basic_symbol< Base >	30
annot::parser::by_kind	
Type access provider for token (enum) based symbols	36
yy::parser::by_kind	
Type access provider for token (enum) based symbols	38
choicenode	41
concatnode	45
annot::parser::context	50
yy::parser::context	51
coordinate	52
driver	54
grammar	56
annot::location	
Two points in a source file	59
yy::location	
Two points in a source file	62
loopnode	66
multinode	70
newlinenode	77
node	80
nodesizes	95
nontermnode	97
nullnode	103
annot::parser	
A Bison parser	105
yy::parser	
A Bison parser	112
annot::position	
A point in a source file	120
yy::position	
A point in a source file	123
productionnode	126
railnode	130
rownode	135

annot::parser::semantic_type	138
singlenode	142
annot::parser::stack< T, S >::slice	
Present a slice of the top of a stack	148
yy::parser::stack< T, S >::slice	
Present a slice of the top of a stack	149
annot::parser::symbol_kind	
Symbol kinds	150
yy::parser::symbol_kind	
Symbol kinds	151
annot::parser::symbol_type	
"External" symbols: returned by the scanner	153
yy::parser::symbol_type	
"External" symbols: returned by the scanner	155
annot::parser::syntax_error	
Syntax errors thrown from user actions	158
yy::parser::syntax_error	
Syntax errors thrown from user actions	160
termnode	162
annot::parser::token	
Token kinds	164
yy::parser::token	
Token kinds	165
yy::parser::value_type	166
yy_buffer_state	170
yy_trans_info	173

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

annot_lexer.cc	175
annot_lexer.hh	208
annot_location.hh	227
annot_parser.cc	229
annot_parser.hh	233
driver.cc	238
driver.hh	239
graph.cc	240
graph.hh	242
lexer.cc	245
lexer.hh	269
location.hh	284
main.cc	286
nodesize.hh	288
optimize.cc	288
output.cc	289
parser.cc	290
parser.hh	294
subsume.cc	299
util.cc	300
util.hh	301

Chapter 8

Namespace Documentation

8.1 annot Namespace Reference

Classes

- class [position](#)
A point in a source file.
- class [location](#)
Two points in a source file.
- class [parser](#)
A Bison parser.

Functions

- [position](#) & [operator](#)+= ([position](#) &res, [position::counter_type](#) width)
Add width columns, in place.
- [position](#) [operator](#)+ ([position](#) res, [position::counter_type](#) width)
Add width columns.
- [position](#) & [operator](#)-= ([position](#) &res, [position::counter_type](#) width)
Subtract width columns, in place.
- [position](#) [operator](#)- ([position](#) res, [position::counter_type](#) width)
Subtract width columns.
- template<typename YYChar >
std::basic_ostream< YYChar > & [operator](#)<< (std::basic_ostream< YYChar > &ostr, const [position](#) &pos)
Intercept output stream redirection.
- [location](#) & [operator](#)+= ([location](#) &res, const [location](#) &end)
Join two locations, in place.
- [location](#) [operator](#)+ ([location](#) res, const [location](#) &end)
Join two locations.
- [location](#) & [operator](#)+= ([location](#) &res, [location::counter_type](#) width)
Add width columns to the end position, in place.
- [location](#) [operator](#)+ ([location](#) res, [location::counter_type](#) width)
Add width columns to the end position.
- [location](#) & [operator](#)-= ([location](#) &res, [location::counter_type](#) width)
Subtract width columns to the end position, in place.
- [location](#) [operator](#)- ([location](#) res, [location::counter_type](#) width)
Subtract width columns to the end position.
- template<typename YYChar >
std::basic_ostream< YYChar > & [operator](#)<< (std::basic_ostream< YYChar > &ostr, const [location](#) &loc)
Intercept output stream redirection.

8.1.1 Function Documentation

8.1.1.1 operator+() [1/3]

```
location annot::operator+ (
    location res,
    const location & end ) [inline]
```

Join two locations.

8.1.1.2 operator+() [2/3]

```
location annot::operator+ (
    location res,
    location::counter_type width ) [inline]
```

Add *width* columns to the end position.

8.1.1.3 operator+() [3/3]

```
position annot::operator+ (
    position res,
    position::counter_type width ) [inline]
```

Add *width* columns.

8.1.1.4 operator+=() [1/3]

```
location& annot::operator+= (
    location & res,
    const location & end ) [inline]
```

Join two locations, in place.

8.1.1.5 operator+=() [2/3]

```
location& annot::operator+= (
    location & res,
    location::counter_type width ) [inline]
```

Add *width* columns to the end position, in place.

8.1.1.6 operator+=() [3/3]

```
position& annot::operator+= (
    position & res,
    position::counter_type width ) [inline]
```

Add *width* columns, in place.

8.1.1.7 operator-() [1/2]

```
location annot::operator- (
    location res,
    location::counter_type width ) [inline]
```

Subtract *width* columns to the end position.

8.1.1.8 operator-() [2/2]

```
position annot::operator- (
    position res,
    position::counter_type width ) [inline]
```

Subtract *width* columns.

8.1.1.9 operator-=() [1/2]

```
location& annot::operator-= (
    location & res,
    location::counter_type width ) [inline]
```

Subtract *width* columns to the end position, in place.

8.1.1.10 operator-=() [2/2]

```
position& annot::operator-= (
    position & res,
    position::counter_type width ) [inline]
```

Subtract *width* columns, in place.

8.1.1.11 operator<<() [1/2]

```
template<typename YYChar >
std::basic_ostream<YYChar>& annot::operator<< (
    std::basic_ostream< YYChar > & ostr,
    const location & loc )
```

Intercept output stream redirection.

Parameters

<i>ostr</i>	the destination output stream
<i>loc</i>	a reference to the location to redirect

Avoid duplicate information.

8.1.1.12 operator<<() [2/2]

```
template<typename YYChar >
std::basic_ostream<YYChar>& annot::operator<< (
    std::basic_ostream< YYChar > & ostr,
    const position & pos )
```

Intercept output stream redirection.

Parameters

<i>ostr</i>	the destination output stream
<i>pos</i>	a reference to the position to redirect

8.2 yy Namespace Reference

Classes

- class [position](#)
A point in a source file.
- class [location](#)
Two points in a source file.
- class [parser](#)
A Bison parser.

Functions

- [position](#) & [operator+=](#) ([position](#) &res, [position::counter_type](#) width)
Add width columns, in place.
- [position](#) [operator+](#) ([position](#) res, [position::counter_type](#) width)
Add width columns.
- [position](#) & [operator-=](#) ([position](#) &res, [position::counter_type](#) width)
Subtract width columns, in place.
- [position](#) [operator-](#) ([position](#) res, [position::counter_type](#) width)
Subtract width columns.
- template<typename YYChar >
std::basic_ostream< YYChar > & [operator<<](#) (std::basic_ostream< YYChar > &ostr, const [position](#) &pos)
Intercept output stream redirection.
- [location](#) & [operator+=](#) ([location](#) &res, const [location](#) &end)

- Join two locations, in place.*

 - `location operator+ (location res, const location &end)`

Join two locations.
- `location & operator+= (location &res, location::counter_type width)`

Add width columns to the end position, in place.
- `location operator+ (location res, location::counter_type width)`

Add width columns to the end position.
- `location & operator-= (location &res, location::counter_type width)`

Subtract width columns to the end position, in place.
- `location operator- (location res, location::counter_type width)`

Subtract width columns to the end position.
- `template<typename YYChar > std::basic_ostream< YYChar > & operator<< (std::basic_ostream< YYChar > &ostr, const location &loc)`

Intercept output stream redirection.

8.2.1 Function Documentation

8.2.1.1 operator+() [1/3]

```
location yy::operator+ (
    location res,
    const location & end ) [inline]
```

Join two locations.

8.2.1.2 operator+() [2/3]

```
location yy::operator+ (
    location res,
    location::counter_type width ) [inline]
```

Add *width* columns to the end position.

8.2.1.3 operator+() [3/3]

```
position yy::operator+ (
    position res,
    position::counter_type width ) [inline]
```

Add *width* columns.

8.2.1.4 operator+=() [1/3]

```
location& yy::operator+= (
    location & res,
    const location & end ) [inline]
```

Join two locations, in place.

8.2.1.5 operator+=() [2/3]

```
location& yy::operator+= (
    location & res,
    location::counter_type width ) [inline]
```

Add *width* columns to the end position, in place.

8.2.1.6 operator+=() [3/3]

```
position& yy::operator+= (
    position & res,
    position::counter_type width ) [inline]
```

Add *width* columns, in place.

8.2.1.7 operator-() [1/2]

```
location yy::operator- (
    location res,
    location::counter_type width ) [inline]
```

Subtract *width* columns to the end position.

8.2.1.8 operator-() [2/2]

```
position yy::operator- (
    position res,
    position::counter_type width ) [inline]
```

Subtract *width* columns.

8.2.1.9 operator-=() [1/2]

```
location& yy::operator-= (
    location & res,
    location::counter_type width ) [inline]
```

Subtract *width* columns to the end position, in place.

8.2.1.10 operator-=() [2/2]

```
position& yy::operator-= (
    position & res,
    position::counter_type width ) [inline]
```

Subtract *width* columns, in place.

8.2.1.11 operator<<() [1/2]

```
template<typename YYChar >
std::basic_ostream<YYChar>& yy::operator<< (
    std::basic_ostream< YYChar > & ostr,
    const location & loc )
```

Intercept output stream redirection.

Parameters

<i>ostr</i>	the destination output stream
<i>loc</i>	a reference to the location to redirect

Avoid duplicate information.

8.2.1.12 operator<<() [2/2]

```
template<typename YYChar >
std::basic_ostream<YYChar>& yy::operator<< (
    std::basic_ostream< YYChar > & ostr,
    const position & pos )
```

Intercept output stream redirection.

Parameters

<i>ostr</i>	the destination output stream
<i>pos</i>	a reference to the position to redirect

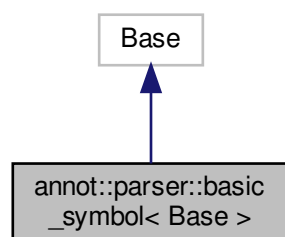
Chapter 9

Class Documentation

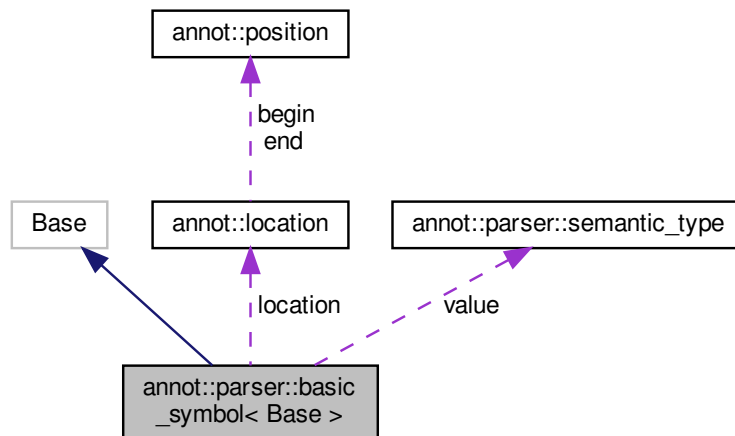
9.1 annot::parser::basic_symbol< Base > Struct Template Reference

```
#include <annot_parser.hh>
```

Inheritance diagram for annot::parser::basic_symbol< Base >:



Collaboration diagram for `annot::parser::basic_symbol< Base >`:



Public Types

- typedef Base `super_type`

Alias to Base.

Public Member Functions

- `basic_symbol ()`
Default constructor.
- `basic_symbol (const basic_symbol &that)`
Copy constructor.
- `basic_symbol (typename Base::kind_type t, const location_type &l)`
Constructors for typed symbols.
- `basic_symbol (typename Base::kind_type t, const annot_t *&v, const location_type &l)`
- `basic_symbol (typename Base::kind_type t, const annotmap *&v, const location_type &l)`
- `basic_symbol (typename Base::kind_type t, const std::string &v, const location_type &l)`
- `~basic_symbol ()`
Destroy the symbol.
- `void clear () YY_NOEXCEPT`
Destroy contents, and record that is empty.
- `const char * name () const YY_NOEXCEPT`
The user-facing name of this symbol.
- `symbol_kind_type type_get () const YY_NOEXCEPT`
Backward compatibility (Bison 3.6).
- `bool empty () const YY_NOEXCEPT`
Whether empty.
- `void move (basic_symbol &s)`
Destructive move, s is emptied into this.

Public Attributes

- [semantic_type](#) value
The semantic value.
- [location_type](#) location
The location.

9.1.1 Detailed Description

```
template<typename Base>
struct annot::parser::basic_symbol< Base >
```

A complete symbol.

Expects its Base type to provide access to the symbol kind via kind ().

Provide access to semantic value and location.

9.1.2 Member Typedef Documentation

9.1.2.1 super_type

```
template<typename Base >
typedef Base annot::parser::basic\_symbol< Base >::super\_type
```

Alias to Base.

9.1.3 Constructor & Destructor Documentation

9.1.3.1 basic_symbol() [1/6]

```
template<typename Base >
annot::parser::basic\_symbol< Base >::basic\_symbol \( \) [inline]
```

Default constructor.

9.1.3.2 basic_symbol() [2/6]

```
template<typename Base >
annot::parser::basic_symbol< Base >::basic_symbol (
    const basic_symbol< Base > & that )
```

Copy constructor.

9.1.3.3 basic_symbol() [3/6]

```
template<typename Base >
annot::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const location_type & l ) [inline]
```

Constructors for typed symbols.

9.1.3.4 basic_symbol() [4/6]

```
template<typename Base >
annot::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const annot_t *& v,
    const location_type & l ) [inline]
```

9.1.3.5 basic_symbol() [5/6]

```
template<typename Base >
annot::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const annotmap *& v,
    const location_type & l ) [inline]
```

9.1.3.6 basic_symbol() [6/6]

```
template<typename Base >
annot::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const std::string & v,
    const location_type & l ) [inline]
```


9.1.3.7 ~basic_symbol()

```
template<typename Base >
annot::parser::basic_symbol< Base >::~~basic_symbol ( ) [inline]
```

Destroy the symbol.

9.1.4 Member Function Documentation

9.1.4.1 clear()

```
template<typename Base >
void annot::parser::basic_symbol< Base >::clear ( ) [inline]
```

Destroy contents, and record that is empty.

9.1.4.2 empty()

```
template<typename Base >
bool annot::parser::basic_symbol< Base >::empty ( ) const
```

Whether empty.

9.1.4.3 move()

```
template<typename Base >
void annot::parser::basic_symbol< Base >::move (
    basic_symbol< Base > & s )
```

Destructive move, *s* is emptied into this.

9.1.4.4 name()

```
template<typename Base >
const char* annot::parser::basic_symbol< Base >::name ( ) const [inline]
```

The user-facing name of this symbol.

9.1.4.5 type_get()

```
template<typename Base >
parser::symbol_kind_type annot::parser::basic_symbol< Base >::type_get ( ) const
```

Backward compatibility (Bison 3.6).

9.1.5 Member Data Documentation

9.1.5.1 location

```
template<typename Base >
location_type annot::parser::basic_symbol< Base >::location
```

The location.

9.1.5.2 value

```
template<typename Base >
semantic_type annot::parser::basic_symbol< Base >::value
```

The semantic value.

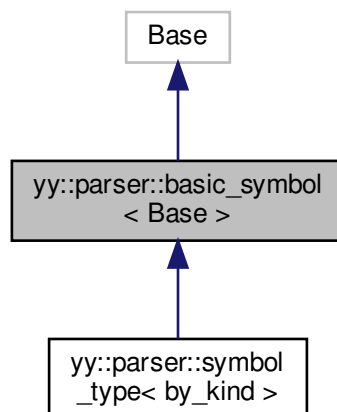
The documentation for this struct was generated from the following file:

- [annot_parser.hh](#)

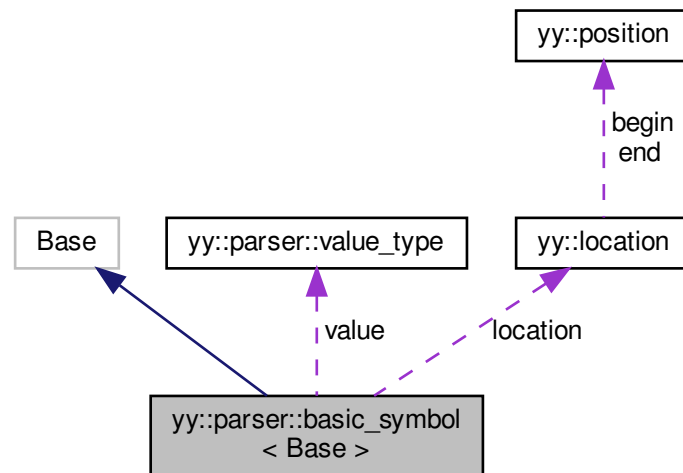
9.2 yy::parser::basic_symbol< Base > Struct Template Reference

```
#include <parser.hh>
```

Inheritance diagram for yy::parser::basic_symbol< Base >:



Collaboration diagram for yy::parser::basic_symbol< Base >:



Public Types

- typedef Base [super_type](#)
Alias to Base.

Public Member Functions

- [basic_symbol](#) () YY_NOEXCEPT
Default constructor.
- [basic_symbol](#) (const [basic_symbol](#) &that)
Copy constructor.
- [basic_symbol](#) (typename Base::kind_type t, const [location_type](#) &l)
Constructors for typed symbols.
- [basic_symbol](#) (typename Base::kind_type t, const [annotmap](#) *&v, const [location_type](#) &l)
- [basic_symbol](#) (typename Base::kind_type t, const [grammar](#) *&v, const [location_type](#) &l)
- [basic_symbol](#) (typename Base::kind_type t, const [node](#) *&v, const [location_type](#) &l)
- [basic_symbol](#) (typename Base::kind_type t, const [productionnode](#) *&v, const [location_type](#) &l)
- [basic_symbol](#) (typename Base::kind_type t, const std::string &v, const [location_type](#) &l)
- [~basic_symbol](#) ()
Destroy the symbol.
- void [clear](#) () YY_NOEXCEPT
Destroy contents, and record that is empty.
- const char * [name](#) () const YY_NOEXCEPT
The user-facing name of this symbol.
- [symbol_kind_type](#) [type_get](#) () const YY_NOEXCEPT
Backward compatibility (Bison 3.6).
- bool [empty](#) () const YY_NOEXCEPT
Whether empty.
- void [move](#) ([basic_symbol](#) &s)
Destructive move, s is emptied into this.

Public Attributes

- [value_type](#) value
The semantic value.
- [location_type](#) location
The location.

9.2.1 Detailed Description

```
template<typename Base>  
struct yy::parser::basic_symbol< Base >
```

A complete symbol.

Expects its Base type to provide access to the symbol kind via `kind ()`.

Provide access to semantic value and location.

9.2.2 Member Typedef Documentation

9.2.2.1 `super_type`

```
template<typename Base >  
typedef Base yy::parser::basic_symbol< Base >::super_type
```

Alias to Base.

9.2.3 Constructor & Destructor Documentation

9.2.3.1 `basic_symbol()` [1/8]

```
template<typename Base >  
yy::parser::basic_symbol< Base >::basic_symbol ( ) [inline]
```

Default constructor.

9.2.3.2 basic_symbol() [2/8]

```
template<typename Base >
yy::parser::basic_symbol< Base >::basic_symbol (
    const basic_symbol< Base > & that )
```

Copy constructor.

9.2.3.3 basic_symbol() [3/8]

```
template<typename Base >
yy::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const location_type & l ) [inline]
```

Constructors for typed symbols.

9.2.3.4 basic_symbol() [4/8]

```
template<typename Base >
yy::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const annotmap *& v,
    const location_type & l ) [inline]
```

9.2.3.5 basic_symbol() [5/8]

```
template<typename Base >
yy::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const grammar *& v,
    const location_type & l ) [inline]
```

9.2.3.6 basic_symbol() [6/8]

```
template<typename Base >
yy::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const node *& v,
    const location_type & l ) [inline]
```

9.2.3.7 basic_symbol() [7/8]

```
template<typename Base >
yy::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const productionnode *& v,
    const location_type & l ) [inline]
```

9.2.3.8 basic_symbol() [8/8]

```
template<typename Base >
yy::parser::basic_symbol< Base >::basic_symbol (
    typename Base::kind_type t,
    const std::string & v,
    const location_type & l ) [inline]
```

9.2.3.9 ~basic_symbol()

```
template<typename Base >
yy::parser::basic_symbol< Base >::~~basic_symbol ( ) [inline]
```

Destroy the symbol.

9.2.4 Member Function Documentation

9.2.4.1 clear()

```
template<typename Base >
void yy::parser::basic_symbol< Base >::clear ( ) [inline]
```

Destroy contents, and record that is empty.

9.2.4.2 empty()

```
template<typename Base >
bool yy::parser::basic_symbol< Base >::empty ( ) const
```

Whether empty.

9.2.4.3 move()

```
template<typename Base >
void yy::parser::basic_symbol< Base >::move (
    basic_symbol< Base > & s )
```

Destructive move, s is emptied into this.

9.2.4.4 name()

```
template<typename Base >
const char* yy::parser::basic_symbol< Base >::name ( ) const [inline]
```

The user-facing name of this symbol.

9.2.4.5 type_get()

```
template<typename Base >
parser::symbol_kind_type yy::parser::basic_symbol< Base >::type_get ( ) const
```

Backward compatibility (Bison 3.6).

9.2.5 Member Data Documentation

9.2.5.1 location

```
template<typename Base >
location_type yy::parser::basic_symbol< Base >::location
```

The location.

9.2.5.2 value

```
template<typename Base >
value_type yy::parser::basic_symbol< Base >::value
```

The semantic value.

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

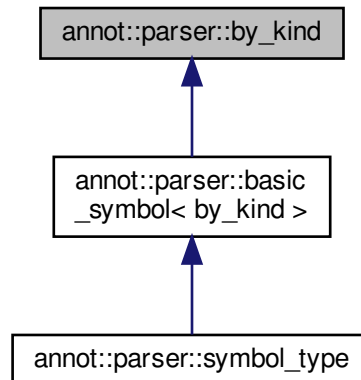
- [parser.hh](#)

9.3 annot::parser::by_kind Struct Reference

Type access provider for token (enum) based symbols.

```
#include <annot_parser.hh>
```

Inheritance diagram for annot::parser::by_kind:



Public Types

- typedef [token_kind_type](#) [kind_type](#)
The symbol kind as needed by the constructor.

Public Member Functions

- [by_kind](#) ()
Default constructor.
- [by_kind](#) (const [by_kind](#) &that)
Copy constructor.
- [by_kind](#) ([kind_type](#) t)
Constructor from (external) token numbers.
- void [clear](#) () [YY_NOEXCEPT](#)
Record that this symbol is empty.
- void [move](#) ([by_kind](#) &that)
Steal the symbol kind from that.
- [symbol_kind_type](#) [kind](#) () const [YY_NOEXCEPT](#)
- [symbol_kind_type](#) [type_get](#) () const [YY_NOEXCEPT](#)
Backward compatibility (Bison 3.6).

Public Attributes

- [symbol_kind_type](#) [kind_](#)

9.3.1 Detailed Description

Type access provider for token (enum) based symbols.

9.3.2 Member Typedef Documentation

9.3.2.1 kind_type

```
typedef token_kind_type annot::parser::by_kind::kind_type
```

The symbol kind as needed by the constructor.

9.3.3 Constructor & Destructor Documentation

9.3.3.1 by_kind() [1/3]

```
annot::parser::by_kind::by_kind ( ) [inline]
```

Default constructor.

9.3.3.2 by_kind() [2/3]

```
annot::parser::by_kind::by_kind (
    const by_kind & that ) [inline]
```

Copy constructor.

9.3.3.3 by_kind() [3/3]

```
annot::parser::by_kind::by_kind (
    kind_type t ) [inline]
```

Constructor from (external) token numbers.

9.3.4 Member Function Documentation

9.3.4.1 clear()

```
void annot::parser::by_kind::clear ( ) [inline]
```

Record that this symbol is empty.

9.3.4.2 kind()

```
parser::symbol_kind_type annot::parser::by_kind::kind ( ) const [inline]
```

The (internal) type number (corresponding to *type*). *empty* when empty.

9.3.4.3 move()

```
void annot::parser::by_kind::move (
    by_kind & that ) [inline]
```

Steal the symbol kind from *that*.

9.3.4.4 type_get()

```
parser::symbol_kind_type annot::parser::by_kind::type_get ( ) const [inline]
```

Backward compatibility (Bison 3.6).

9.3.5 Member Data Documentation

9.3.5.1 kind_

```
symbol_kind_type annot::parser::by_kind::kind_
```

The symbol kind. *S_YYEMPTY* when empty.

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

- [annot_parser.hh](#)

9.4 yy::parser::by_kind Struct Reference

Type access provider for token (enum) based symbols.

```
#include <parser.hh>
```

Public Types

- typedef [token_kind_type](#) [kind_type](#)
The symbol kind as needed by the constructor.

Public Member Functions

- [by_kind](#) () [YY_NOEXCEPT](#)
Default constructor.
- [by_kind](#) (const [by_kind](#) &that) [YY_NOEXCEPT](#)
Copy constructor.
- [by_kind](#) ([kind_type](#) t) [YY_NOEXCEPT](#)
Constructor from (external) token numbers.
- void [clear](#) () [YY_NOEXCEPT](#)
Record that this symbol is empty.
- void [move](#) ([by_kind](#) &that)
Steal the symbol kind from that.
- [symbol_kind_type](#) [kind](#) () const [YY_NOEXCEPT](#)
- [symbol_kind_type](#) [type_get](#) () const [YY_NOEXCEPT](#)
Backward compatibility (Bison 3.6).

Public Attributes

- [symbol_kind_type](#) [kind_](#)

9.4.1 Detailed Description

Type access provider for token (enum) based symbols.

9.4.2 Member Typedef Documentation

9.4.2.1 [kind_type](#)

```
typedef token\_kind\_type yy::parser::by\_kind::kind\_type
```

The symbol kind as needed by the constructor.

9.4.3 Constructor & Destructor Documentation

9.4.3.1 `by_kind()` [1/3]

```
yy::parser::by_kind::by_kind ( ) [inline]
```

Default constructor.

9.4.3.2 `by_kind()` [2/3]

```
yy::parser::by_kind::by_kind (
    const by_kind & that ) [inline]
```

Copy constructor.

9.4.3.3 `by_kind()` [3/3]

```
yy::parser::by_kind::by_kind (
    kind_type t ) [inline]
```

Constructor from (external) token numbers.

9.4.4 Member Function Documentation

9.4.4.1 `clear()`

```
void yy::parser::by_kind::clear ( ) [inline]
```

Record that this symbol is empty.

9.4.4.2 `kind()`

```
parser::symbol_kind_type yy::parser::by_kind::kind ( ) const [inline]
```

The (internal) type number (corresponding to *type*). *empty* when empty.

9.4.4.3 `move()`

```
void yy::parser::by_kind::move (
    by_kind & that ) [inline]
```

Steal the symbol kind from *that*.

9.4.4.4 type_get()

```
parser::symbol_kind_type yy::parser::by_kind::type_get ( ) const [inline]
```

Backward compatibility (Bison 3.6).

9.4.5 Member Data Documentation

9.4.5.1 kind_

```
symbol_kind_type yy::parser::by_kind::kind_
```

The symbol kind. *S_YYEMPTY* when empty.

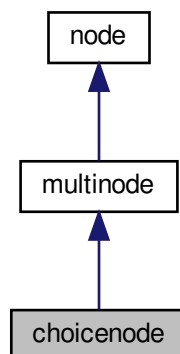
The documentation for this struct was generated from the following file:

- [parser.hh](#)

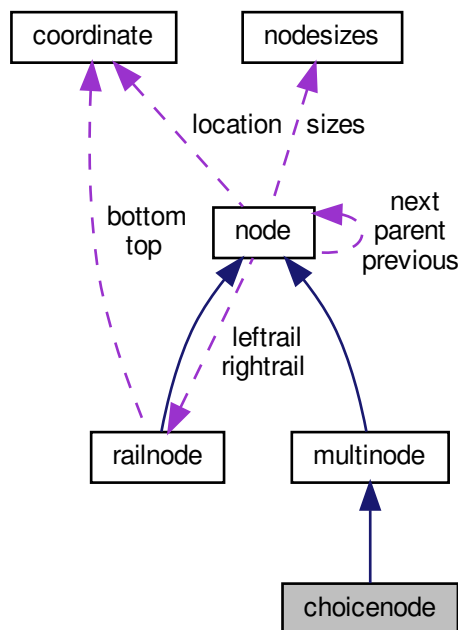
9.5 choicenode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for choicenode:



Collaboration diagram for choicenode:



Public Member Functions

- [choicenode](#) ([node](#) *p)
- [choicenode](#) (const [choicenode](#) &original)
- virtual [choicenode](#) * [clone](#) () const
- virtual void [insert](#) ([node](#) *[node](#))
- virtual [~choicenode](#) ()
- virtual int [rail_left](#) ()
- virtual int [rail_right](#) ()
- virtual void [drawToLeftRail](#) (ofstream &outs, [railnode](#) *p, [vraildir](#) join, int drawself)
- virtual void [drawToRightRail](#) (ofstream &outs, [railnode](#) *p, [vraildir](#) join, int drawself)
- virtual void [dump](#) (int depth) const
- virtual int [mergeChoices](#) (int depth)
- virtual void [fixSkips](#) ()
- virtual string [texName](#) ()

Additional Inherited Members

9.5.1 Constructor & Destructor Documentation

9.5.1.1 choicenode() [1/2]

```
choicenode::choicenode (
    node * p )
```

9.5.1.2 choicenode() [2/2]

```
choicenode::choicenode (
    const choicenode & original )
```

9.5.1.3 ~choicenode()

```
virtual choicenode::~~choicenode ( ) [inline], [virtual]
```

9.5.2 Member Function Documentation

9.5.2.1 clone()

```
choicenode * choicenode::clone ( ) const [virtual]
```

Reimplemented from [multinode](#).

9.5.2.2 drawToLeftRail()

```
void choicenode::drawToLeftRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.5.2.3 drawToRightRail()

```
void choicenode::drawToRightRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.5.2.4 dump()

```
void choicenode::dump (
    int depth ) const [virtual]
```

Reimplemented from [node](#).

9.5.2.5 fixSkips()

```
void choicenode::fixSkips ( ) [virtual]
```

Reimplemented from [multinode](#).

9.5.2.6 insert()

```
void choicenode::insert (
    node * node ) [virtual]
```

Reimplemented from [multinode](#).

9.5.2.7 mergeChoices()

```
int choicenode::mergeChoices (
    int depth ) [virtual]
```

Reimplemented from [multinode](#).

9.5.2.8 rail_left()

```
virtual int choicenode::rail_left ( ) [inline], [virtual]
```

9.5.2.9 rail_right()

```
virtual int choicenode::rail_right ( ) [inline], [virtual]
```

9.5.2.10 texName()

```
virtual string choicenode::texName ( ) [inline], [virtual]
```

Reimplemented from [multinode](#).

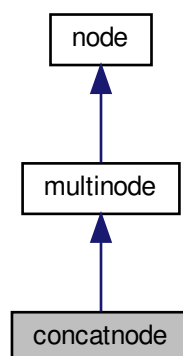
The documentation for this class was generated from the following files:

- [graph.hh](#)
- [graph.cc](#)
- [optimize.cc](#)
- [output.cc](#)
- [subsume.cc](#)

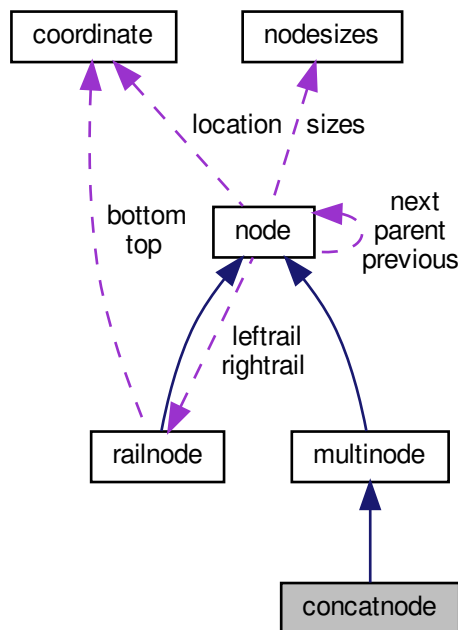
9.6 concatnode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for concatnode:



Collaboration diagram for concatnode:



Public Member Functions

- [concatnode](#) ([node](#) *p)
- [concatnode](#) (const [concatnode](#) &original)
- virtual [concatnode](#) * [clone](#) () const
- virtual [~concatnode](#) ()
- virtual void [dump](#) (int depth) const
- virtual void [insert](#) ([node](#) *p)
- virtual [coordinate](#) [place](#) (ofstream &outs, int draw, int drawrails, [coordinate](#) start, [node](#) *parent, int depth)
- virtual int [mergeConcats](#) (int depth)
- virtual int [analyzeOptLoops](#) (int depth)
- virtual int [analyzeNonOptLoops](#) (int depth)
- virtual void [drawToLeftRail](#) (ofstream &outs, [railnode](#) *p, [vraildir](#) join, int drawself)
- virtual void [drawToRightRail](#) (ofstream &outs, [railnode](#) *p, [vraildir](#) join, int drawself)
- virtual void [mergeRails](#) ()
- virtual void [setPrevious](#) ([node](#) *p)
- virtual void [setNext](#) ([node](#) *p)
- virtual void [fixSkips](#) ()
- virtual [node](#) * [createRows](#) ()

Additional Inherited Members

9.6.1 Constructor & Destructor Documentation

9.6.1.1 concatnode() [1/2]

```
concatnode::concatnode (
    node * p )
```

9.6.1.2 concatnode() [2/2]

```
concatnode::concatnode (
    const concatnode & original )
```

9.6.1.3 ~concatnode()

```
virtual concatnode::~~concatnode ( ) [inline], [virtual]
```

9.6.2 Member Function Documentation

9.6.2.1 analyzeNonOptLoops()

```
int concatnode::analyzeNonOptLoops (
    int depth ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.2 analyzeOptLoops()

```
int concatnode::analyzeOptLoops (
    int depth ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.3 clone()

```
concatnode * concatnode::clone ( ) const [virtual]
```

Reimplemented from [multinode](#).

9.6.2.4 createRows()

```
node * concatnode::createRows ( ) [virtual]
```

Reimplemented from [node](#).

9.6.2.5 drawToLeftRail()

```
void concatnode::drawToLeftRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.6.2.6 drawToRightRail()

```
void concatnode::drawToRightRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.6.2.7 dump()

```
void concatnode::dump (
    int depth ) const [virtual]
```

Reimplemented from [node](#).

9.6.2.8 fixSkips()

```
void concatnode::fixSkips ( ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.9 insert()

```
void concatnode::insert (
    node * p ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.10 mergeConcats()

```
int concatnode::mergeConcats (
    int depth ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.11 mergeRails()

```
void concatnode::mergeRails ( ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.12 place()

```
coordinate concatnode::place (
    ofstream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.13 setNext()

```
void concatnode::setNext (
    node * p ) [virtual]
```

Reimplemented from [multinode](#).

9.6.2.14 setPrevious()

```
void concatnode::setPrevious (
    node * p ) [virtual]
```

Reimplemented from [multinode](#).

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

- [graph.hh](#)
- [graph.cc](#)
- [optimize.cc](#)
- [output.cc](#)
- [subsume.cc](#)

9.7 annot::parser::context Class Reference

```
#include <annot_parser.hh>
```

Public Member Functions

- [context](#) (const [parser](#) & yyparser, const [symbol_type](#) & yyla)
- const [symbol_type](#) & [lookahead](#) () const [YY_NOEXCEPT](#)
- [symbol_kind_type](#) [token](#) () const [YY_NOEXCEPT](#)
- const [location_type](#) & [location](#) () const [YY_NOEXCEPT](#)
- int [expected_tokens](#) ([symbol_kind_type](#) yyarg[], int yyargn) const

9.7.1 Constructor & Destructor Documentation

9.7.1.1 context()

```
annot::parser::context::context (
    const parser & yyparser,
    const symbol\_type & yyla )
```

9.7.2 Member Function Documentation

9.7.2.1 expected_tokens()

```
int annot::parser::context::expected_tokens (
    symbol\_kind\_type yyarg[],
    int yyargn ) const
```

Put in YYARG at most YYARGN of the expected tokens, and return the number of tokens stored in YYARG. If YYARG is null, return the number of expected tokens (guaranteed to be less than YYNTOKENS).

9.7.2.2 location()

```
const location_type& annot::parser::context::location ( ) const [inline]
```

9.7.2.3 lookahead()

```
const symbol_type& annot::parser::context::lookahead ( ) const [inline]
```

9.7.2.4 token()

```
symbol_kind_type annot::parser::context::token ( ) const [inline]
```

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

- [annot_parser.hh](#)
- [annot_parser.cc](#)

9.8 yy::parser::context Class Reference

```
#include <parser.hh>
```

Public Member Functions

- [context](#) (const [parser](#) &yyparser, const [symbol_type](#) &yyla)
- const [symbol_type](#) & [lookahead](#) () const [YY_NOEXCEPT](#)
- [symbol_kind_type](#) [token](#) () const [YY_NOEXCEPT](#)
- const [location_type](#) & [location](#) () const [YY_NOEXCEPT](#)
- int [expected_tokens](#) ([symbol_kind_type](#) yyarg[], int yyargn) const

9.8.1 Constructor & Destructor Documentation

9.8.1.1 context()

```
yy::parser::context::context (
    const parser & yyparser,
    const symbol\_type & yylla )
```

9.8.2 Member Function Documentation

9.8.2.1 expected_tokens()

```
int yy::parser::context::expected_tokens (
    symbol_kind_type yyarg[],
    int yyargn ) const
```

Put in YYARG at most YYARGN of the expected tokens, and return the number of tokens stored in YYARG. If YYARG is null, return the number of expected tokens (guaranteed to be less than YYNTOKENS).

9.8.2.2 location()

```
const location_type& yy::parser::context::location ( ) const [inline]
```

9.8.2.3 lookahead()

```
const symbol_type& yy::parser::context::lookahead ( ) const [inline]
```

9.8.2.4 token()

```
symbol_kind_type yy::parser::context::token ( ) const [inline]
```

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

- [parser.hh](#)
- [parser.cc](#)

9.9 coordinate Class Reference

```
#include <nodesize.hh>
```

Public Member Functions

- [coordinate](#) ()
- [coordinate](#) (float nx, float ny)
- [coordinate operator+](#) ([coordinate](#) r)
- [coordinate operator-](#) ([coordinate](#) r)
- [coordinate & operator=](#) ([coordinate](#) r)

Public Attributes

- float `x`
- float `y`

Friends

- ostream & `operator<<` (std::ostream &out, const `coordinate` &c)

9.9.1 Constructor & Destructor Documentation

9.9.1.1 `coordinate()` [1/2]

```
coordinate::coordinate ( ) [inline]
```

9.9.1.2 `coordinate()` [2/2]

```
coordinate::coordinate (
    float nx,
    float ny ) [inline]
```

9.9.2 Member Function Documentation

9.9.2.1 `operator+()`

```
coordinate coordinate::operator+ (
    coordinate r ) [inline]
```

9.9.2.2 `operator-()`

```
coordinate coordinate::operator- (
    coordinate r ) [inline]
```

9.9.2.3 operator=()

```
coordinate& coordinate::operator= (
    coordinate r ) [inline]
```

9.9.3 Friends And Related Function Documentation

9.9.3.1 operator<<

```
ostream& operator<< (
    std::ostream & out,
    const coordinate & c ) [friend]
```

9.9.4 Member Data Documentation

9.9.4.1 x

```
float coordinate::x
```

9.9.4.2 y

```
float coordinate::y
```

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

- [nodesize.hh](#)

9.10 driver Class Reference

```
#include <driver.hh>
```

Public Member Functions

- [driver](#) (ofstream *out)
- ofstream & [outs](#) ()
- [node](#) * [addTerminal](#) (string &s)
- [node](#) * [addString](#) (string &s)
- int [parse](#) (const char *f, int opt, int fig)
- void [scan_begin](#) ()
- void [scan_begin](#) (stringstream &s)
- void [scan_end](#) ()
- int [get_result](#) ()
- [yy::location](#) & [get_location](#) ()

9.10.1 Constructor & Destructor Documentation

9.10.1.1 driver()

```
driver::driver (
    ofstream * out )
```

9.10.2 Member Function Documentation

9.10.2.1 addString()

```
node* driver::addString (
    string & s ) [inline]
```

9.10.2.2 addTerminal()

```
node* driver::addTerminal (
    string & s ) [inline]
```

9.10.2.3 get_location()

```
yy::location& driver::get_location ( ) [inline]
```

9.10.2.4 get_result()

```
int driver::get_result ( ) [inline]
```

9.10.2.5 outs()

```
ofstream& driver::outs ( ) [inline]
```

9.10.2.6 parse()

```
int driver::parse (
    const char * f,
    int opt,
    int fig )
```

9.10.2.7 scan_begin() [1/2]

```
void driver::scan_begin ( )
```

9.10.2.8 scan_begin() [2/2]

```
void driver::scan_begin (
    stringstream & s )
```

9.10.2.9 scan_end()

```
void driver::scan_end ( )
```

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

- [driver.hh](#)
- [driver.cc](#)
- [lexer.cc](#)

9.11 grammar Class Reference

```
#include <graph.hh>
```

Public Member Functions

- [grammar](#) ([node](#) **p*)
- [~grammar](#) ()
- void [insert](#) ([productionnode](#) **node*)
- void [dump](#) () const
- void [optimize](#) ()
- void [subsume](#) ()
- void [place](#) (ofstream &*outs*)
- void [mergeRails](#) ()
- void [setParent](#) ()
- void [setPrevious](#) ()
- void [setNext](#) ()
- void [fixSkips](#) ()
- void [createRows](#) ()

9.11.1 Constructor & Destructor Documentation

9.11.1.1 grammar()

```
grammar::grammar (
    node * p ) [inline]
```

9.11.1.2 ~grammar()

```
grammar::~~grammar ( )
```

9.11.2 Member Function Documentation

9.11.2.1 createRows()

```
void grammar::createRows ( )
```

9.11.2.2 dump()

```
void grammar::dump ( ) const
```

9.11.2.3 fixSkips()

```
void grammar::fixSkips ( ) [inline]
```

9.11.2.4 insert()

```
void grammar::insert (
    productionnode * node ) [inline]
```

9.11.2.5 mergeRails()

```
void grammar::mergeRails ( )
```

9.11.2.6 optimize()

```
void grammar::optimize ( )
```

9.11.2.7 place()

```
void grammar::place (
    ostream & outs )
```

9.11.2.8 setNext()

```
void grammar::setNext ( )
```

9.11.2.9 setParent()

```
void grammar::setParent ( )
```

9.11.2.10 setPrevious()

```
void grammar::setPrevious ( )
```

9.11.2.11 subsume()

```
void grammar::subsume ( )
```

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

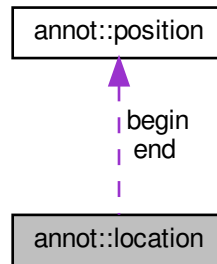
- [graph.hh](#)
- [graph.cc](#)
- [optimize.cc](#)
- [output.cc](#)
- [subsume.cc](#)

9.12 annot::location Class Reference

Two points in a source file.

```
#include <annot_location.hh>
```

Collaboration diagram for annot::location:



Public Types

- typedef [position::filename_type](#) filename_type
Type for file name.
- typedef [position::counter_type](#) counter_type
Type for line and column numbers.

Public Member Functions

- [location](#) (const [position](#) &b, const [position](#) &e)
Construct a location from b to e.
- [location](#) (const [position](#) &p=[position](#)())
Construct a 0-width location in p.
- [location](#) ([filename_type](#) *f, [counter_type](#) l=1, [counter_type](#) c=1)
Construct a 0-width location in f, l, c.
- void [initialize](#) ([filename_type](#) *f=YY_NULLPTR, [counter_type](#) l=1, [counter_type](#) c=1)
Initialization.

Line and Column related manipulators

- void [step](#) ()
Reset initial location to final location.
- void [columns](#) ([counter_type](#) count=1)
Extend the current location to the COUNT next columns.
- void [lines](#) ([counter_type](#) count=1)
Extend the current location to the COUNT next lines.

Public Attributes

- [position begin](#)
Beginning of the located region.
- [position end](#)
End of the located region.

9.12.1 Detailed Description

Two points in a source file.

9.12.2 Member Typedef Documentation

9.12.2.1 counter_type

```
typedef position::counter\_type annot::location::counter\_type
```

Type for line and column numbers.

9.12.2.2 filename_type

```
typedef position::filename\_type annot::location::filename\_type
```

Type for file name.

9.12.3 Constructor & Destructor Documentation

9.12.3.1 location() [1/3]

```
annot::location::location (  
    const position & b,  
    const position & e ) [inline]
```

Construct a location from *b* to *e*.

9.12.3.2 location() [2/3]

```
annot::location::location (
    const position & p = position () ) [inline], [explicit]
```

Construct a 0-width location in *p*.

9.12.3.3 location() [3/3]

```
annot::location::location (
    filename_type * f,
    counter_type l = 1,
    counter_type c = 1 ) [inline], [explicit]
```

Construct a 0-width location in *f*, *l*, *c*.

9.12.4 Member Function Documentation

9.12.4.1 columns()

```
void annot::location::columns (
    counter_type count = 1 ) [inline]
```

Extend the current location to the COUNT next columns.

9.12.4.2 initialize()

```
void annot::location::initialize (
    filename_type * f = YY_NULLPTR,
    counter_type l = 1,
    counter_type c = 1 ) [inline]
```

Initialization.

9.12.4.3 lines()

```
void annot::location::lines (
    counter_type count = 1 ) [inline]
```

Extend the current location to the COUNT next lines.

9.12.4.4 step()

```
void annot::location::step ( ) [inline]
```

Reset initial location to final location.

9.12.5 Member Data Documentation

9.12.5.1 begin

```
position annot::location::begin
```

Beginning of the located region.

9.12.5.2 end

```
position annot::location::end
```

End of the located region.

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

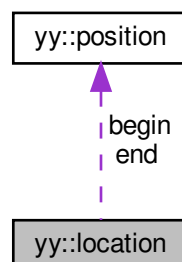
- [annot_location.hh](#)

9.13 yy::location Class Reference

Two points in a source file.

```
#include <location.hh>
```

Collaboration diagram for yy::location:



Public Types

- typedef [position::filename_type](#) [filename_type](#)
Type for file name.
- typedef [position::counter_type](#) [counter_type](#)
Type for line and column numbers.

Public Member Functions

- [location](#) (const [position](#) &b, const [position](#) &e)
Construct a location from b to e.
- [location](#) (const [position](#) &p=[position](#)())
Construct a 0-width location in p.
- [location](#) ([filename_type](#) *f, [counter_type](#) l=1, [counter_type](#) c=1)
Construct a 0-width location in f, l, c.
- void [initialize](#) ([filename_type](#) *f=YY_NULLPTR, [counter_type](#) l=1, [counter_type](#) c=1)
Initialization.

Line and Column related manipulators

- void [step](#) ()
Reset initial location to final location.
- void [columns](#) ([counter_type](#) count=1)
Extend the current location to the COUNT next columns.
- void [lines](#) ([counter_type](#) count=1)
Extend the current location to the COUNT next lines.

Public Attributes

- [position begin](#)
Beginning of the located region.
- [position end](#)
End of the located region.

9.13.1 Detailed Description

Two points in a source file.

9.13.2 Member Typedef Documentation

9.13.2.1 counter_type

```
typedef position::counter\_type yy::location::counter\_type
```

Type for line and column numbers.

9.13.2.2 filename_type

```
typedef position::filename_type yy::location::filename_type
```

Type for file name.

9.13.3 Constructor & Destructor Documentation

9.13.3.1 location() [1/3]

```
yy::location::location (
    const position & b,
    const position & e ) [inline]
```

Construct a location from *b* to *e*.

9.13.3.2 location() [2/3]

```
yy::location::location (
    const position & p = position () ) [inline], [explicit]
```

Construct a 0-width location in *p*.

9.13.3.3 location() [3/3]

```
yy::location::location (
    filename_type * f,
    counter_type l = 1,
    counter_type c = 1 ) [inline], [explicit]
```

Construct a 0-width location in *f*, *l*, *c*.

9.13.4 Member Function Documentation

9.13.4.1 columns()

```
void yy::location::columns (
    counter_type count = 1 ) [inline]
```

Extend the current location to the COUNT next columns.

9.13.4.2 initialize()

```
void yy::location::initialize (
    filename_type * f = YY_NULLPTR,
    counter_type l = 1,
    counter_type c = 1 ) [inline]
```

Initialization.

9.13.4.3 lines()

```
void yy::location::lines (
    counter_type count = 1 ) [inline]
```

Extend the current location to the COUNT next lines.

9.13.4.4 step()

```
void yy::location::step ( ) [inline]
```

Reset initial location to final location.

9.13.5 Member Data Documentation

9.13.5.1 begin

```
position yy::location::begin
```

Beginning of the located region.

9.13.5.2 end

```
position yy::location::end
```

End of the located region.

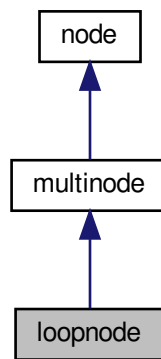
The documentation for this class was generated from the following file:

- [location.hh](#)

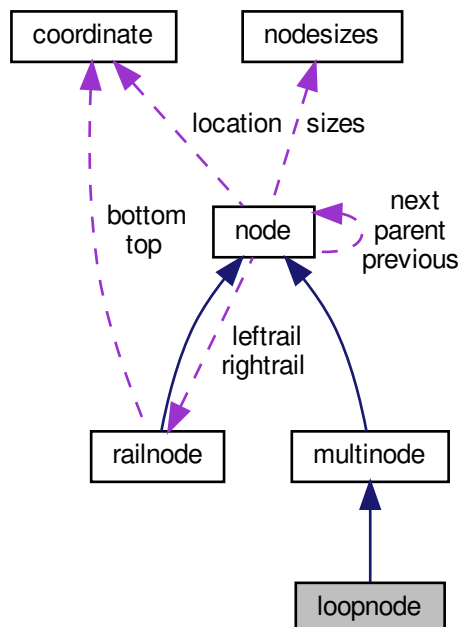
9.14 loopnode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for loopnode:



Collaboration diagram for loopnode:



Public Member Functions

- `loopnode` (`node *node`)
- `loopnode` (`const loopnode &original`)
- `loopnode * clone` () const
- virtual `~loopnode` ()
- virtual void `dump` (int depth) const
- virtual void `drawToLeftRail` (ofstream &outs, `railnode *p`, `vraildir` join, int drawself)
- virtual void `drawToRightRail` (ofstream &outs, `railnode *p`, `vraildir` join, int drawself)
- `node * getRepeat` ()
- void `setRepeat` (`node *r`)
- `node * getBody` ()
- void `setBody` (`node *r`)
- virtual void `fixSkips` ()
- virtual string `texName` ()

Additional Inherited Members

9.14.1 Constructor & Destructor Documentation

9.14.1.1 `loopnode()` [1/2]

```
loopnode::loopnode (
    node * node )
```

9.14.1.2 `loopnode()` [2/2]

```
loopnode::loopnode (
    const loopnode & original )
```

9.14.1.3 `~loopnode()`

```
virtual loopnode::~~loopnode ( ) [inline], [virtual]
```

9.14.2 Member Function Documentation

9.14.2.1 clone()

```
loopnode * loopnode::clone ( ) const [virtual]
```

Reimplemented from [multinode](#).

9.14.2.2 drawToLeftRail()

```
void loopnode::drawToLeftRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.14.2.3 drawToRightRail()

```
void loopnode::drawToRightRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.14.2.4 dump()

```
void loopnode::dump (
    int depth ) const [virtual]
```

Reimplemented from [node](#).

9.14.2.5 fixSkips()

```
void loopnode::fixSkips ( ) [virtual]
```

Reimplemented from [multinode](#).

9.14.2.6 `getBody()`

```
node * loopnode::getBody ( )
```

9.14.2.7 `getRepeat()`

```
node * loopnode::getRepeat ( )
```

9.14.2.8 `setBody()`

```
void loopnode::setBody (
    node * r )
```

9.14.2.9 `setRepeat()`

```
void loopnode::setRepeat (
    node * r )
```

9.14.2.10 `texName()`

```
virtual string loopnode::texName ( ) [inline], [virtual]
```

Reimplemented from [multinode](#).

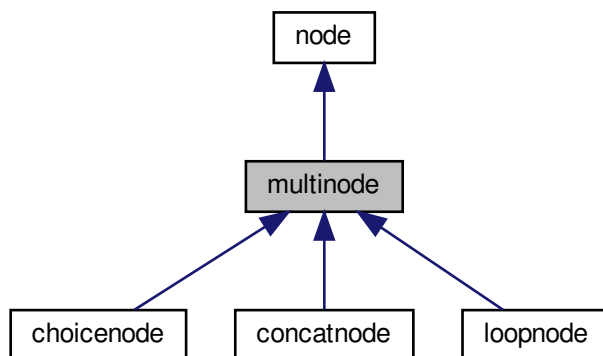
The documentation for this class was generated from the following files:

- [graph.hh](#)
- [graph.cc](#)
- [output.cc](#)
- [subsume.cc](#)

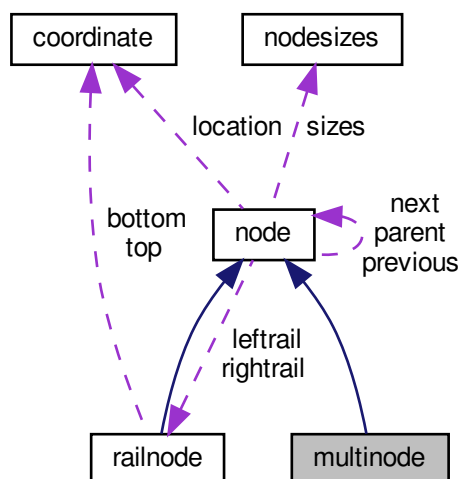
9.15 multinode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for multinode:



Collaboration diagram for multinode:



Public Member Functions

- `multinode` (`node *p`)
- `multinode` (`const multinode &original`)

- virtual `multinode * clone ()` const
- virtual void `forgetChild (int n)`
- virtual `~multinode ()`
- virtual void `mergeRails ()`
- virtual void `insert (node *node)`
- virtual void `insertFirst (node *node)`
- virtual int `numChildren ()`
- virtual `node * getChild (int n)`
- virtual `coordinate place` (ofstream &outs, int draw, int drawrails, `coordinate` start, `node *parent`, int depth)
- virtual int `operator== (node &r)`
- virtual int `operator!= (node &r)`
- virtual `node * subsume` (regex_t *name, `node *replacement`)
- virtual void `setParent (node *p)`
- virtual void `setPrevious (node *p)`
- virtual void `setNext (node *p)`
- virtual int `liftConcats (int depth)`
- virtual int `mergeConcats (int depth)`
- virtual int `mergeChoices (int depth)`
- virtual int `analyzeOptLoops (int depth)`
- virtual int `analyzeNonOptLoops (int depth)`
- virtual void `fixSkips ()`
- virtual string `texName ()`

Protected Attributes

- vector< `node *` > `nodes`

Friends

- class `concatnode`

Additional Inherited Members

9.15.1 Constructor & Destructor Documentation

9.15.1.1 `multinode()` [1/2]

```
multinode::multinode (
    node * p )
```

9.15.1.2 `multinode()` [2/2]

```
multinode::multinode (
    const multinode & original )
```

9.15.1.3 ~multinode()

```
multinode::~~multinode ( ) [virtual]
```

9.15.2 Member Function Documentation

9.15.2.1 analyzeNonOptLoops()

```
int multinode::analyzeNonOptLoops (
    int depth ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#).

9.15.2.2 analyzeOptLoops()

```
int multinode::analyzeOptLoops (
    int depth ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#).

9.15.2.3 clone()

```
multinode * multinode::clone ( ) const [virtual]
```

Implements [node](#).

Reimplemented in [concatnode](#), [loopnode](#), and [choicenode](#).

9.15.2.4 fixSkips()

```
void multinode::fixSkips ( ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#), [loopnode](#), and [choicenode](#).

9.15.2.5 forgetChild()

```
void multinode::forgetChild (
    int n ) [virtual]
```

Reimplemented from [node](#).

9.15.2.6 getChild()

```
virtual node* multinode::getChild (
    int n ) [inline], [virtual]
```

Reimplemented from [node](#).

9.15.2.7 insert()

```
void multinode::insert (
    node * node ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#), and [choicenode](#).

9.15.2.8 insertFirst()

```
void multinode::insertFirst (
    node * node ) [virtual]
```

9.15.2.9 liftConcats()

```
int multinode::liftConcats (
    int depth ) [virtual]
```

Reimplemented from [node](#).

9.15.2.10 mergeChoices()

```
int multinode::mergeChoices (
    int depth ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [choicenode](#).

9.15.2.11 mergeConcats()

```
int multinode::mergeConcats (
    int depth ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#).

9.15.2.12 mergeRails()

```
void multinode::mergeRails ( ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#).

9.15.2.13 numChildren()

```
virtual int multinode::numChildren ( ) [inline], [virtual]
```

Reimplemented from [node](#).

9.15.2.14 operator"!="()

```
virtual int multinode::operator!= (
    node & r ) [inline], [virtual]
```

Reimplemented from [node](#).

9.15.2.15 operator==()

```
int multinode::operator== (
    node & r ) [virtual]
```

Reimplemented from [node](#).

9.15.2.16 place()

```
coordinate multinode::place (
    ofstream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#).

9.15.2.17 setNext()

```
void multinode::setNext (
    node * p ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#).

9.15.2.18 setParent()

```
void multinode::setParent (
    node * p ) [virtual]
```

Reimplemented from [node](#).

9.15.2.19 setPrevious()

```
void multinode::setPrevious (
    node * p ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [concatnode](#).

9.15.2.20 subsume()

```
node * multinode::subsume (
    regex_t * name,
    node * replacement ) [virtual]
```

Reimplemented from [node](#).

9.15.2.21 texName()

```
virtual string multinode::texName ( ) [inline], [virtual]
```

Reimplemented from [node](#).

Reimplemented in [loopnode](#), and [choicenode](#).

9.15.3 Friends And Related Function Documentation

9.15.3.1 concatnode

```
friend class concatnode [friend]
```

9.15.4 Member Data Documentation

9.15.4.1 nodes

```
vector<node*> multinode::nodes [protected]
```

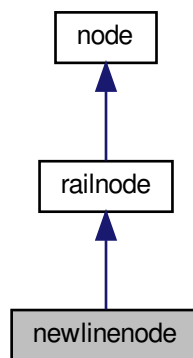
The documentation for this class was generated from the following files:

- [graph.hh](#)
- [graph.cc](#)
- [optimize.cc](#)
- [output.cc](#)
- [subsume.cc](#)

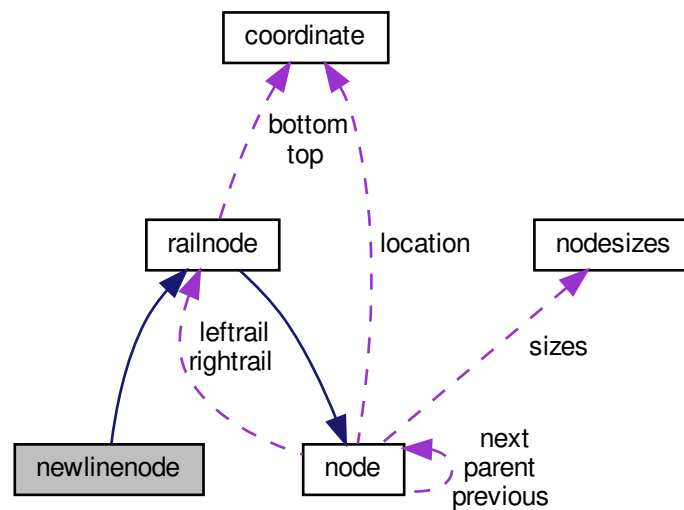
9.16 newlinenode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for newlinenode:



Collaboration diagram for newlinenode:



Public Member Functions

- [newlinenode](#) ()

- [newlinenode](#) (const [newlinenode](#) &original)
- virtual [newlinenode](#) * [clone](#) () const
- virtual [~newlinenode](#) ()
- virtual [coordinate place](#) (ofstream &outs, int draw, int drawrails, [coordinate](#) start, [node](#) *parent, int depth)
- virtual int [rail_left](#) ()
- virtual int [rail_right](#) ()
- virtual void [setLineHeight](#) (float h)
- virtual void [dump](#) (int depth) const

Additional Inherited Members

9.16.1 Constructor & Destructor Documentation

9.16.1.1 newlinenode() [1/2]

```
newlinenode::newlinenode ( )
```

9.16.1.2 newlinenode() [2/2]

```
newlinenode::newlinenode (
    const newlinenode & original )
```

9.16.1.3 ~newlinenode()

```
virtual newlinenode::~~newlinenode ( ) [inline], [virtual]
```

9.16.2 Member Function Documentation

9.16.2.1 clone()

```
newlinenode * newlinenode::clone ( ) const [virtual]
```

Reimplemented from [railnode](#).

9.16.2.2 dump()

```
void newlinenode::dump (
    int depth ) const [virtual]
```

Reimplemented from [railnode](#).

9.16.2.3 place()

```
coordinate newlinenode::place (
    ostream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [railnode](#).

9.16.2.4 rail_left()

```
virtual int newlinenode::rail_left ( ) [inline], [virtual]
```

9.16.2.5 rail_right()

```
virtual int newlinenode::rail_right ( ) [inline], [virtual]
```

9.16.2.6 setLineHeight()

```
virtual void newlinenode::setLineHeight (
    float h ) [inline], [virtual]
```

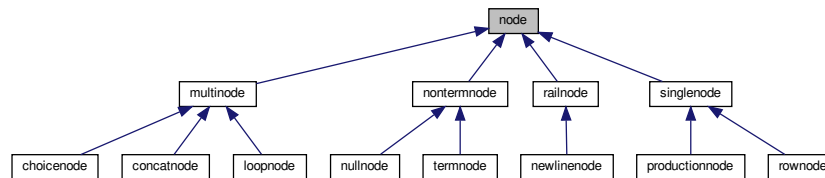
The documentation for this class was generated from the following files:

- [graph.hh](#)
- [graph.cc](#)
- [output.cc](#)

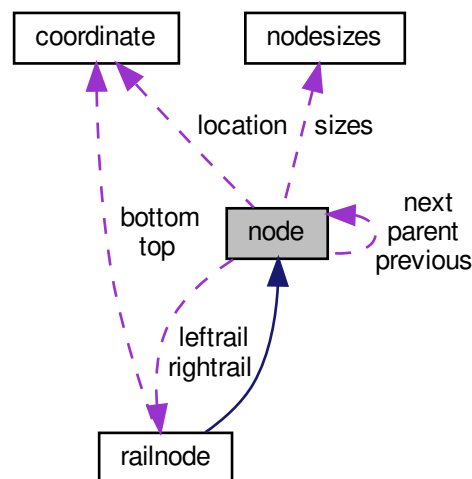
9.17 node Class Reference

```
#include <graph.hh>
```

Inheritance diagram for node:



Collaboration diagram for node:



Public Member Functions

- `node ()`
- `node (const node &original)`
- `virtual node * clone () const =0`
- `virtual ~node ()`
- `virtual void setParent (node *p)`
- `virtual void setPrevious (node *p)`
- `virtual void setNext (node *p)`
- `virtual void setLeftRail (railnode *p)`
- `virtual void setRightRail (railnode *p)`
- `railnode * getLeftRail ()`

- `railnode * getRightRail ()`
- virtual void `drawToLeftRail` (ofstream &outs, `railnode *p`, `vraildir` join, int drawself)
- virtual void `drawToRightRail` (ofstream &outs, `railnode *p`, `vraildir` join, int drawself)
- void `makeDead ()`
- int `isDead ()`
- void `setBeforeSkip` (float s)
- void `setDrawToPrev` (int d)
- float `getBeforeSkip ()`
- int `getDrawToPrev ()`
- `node * getParent ()`
- `node * getNext ()`
- `node * getPrevious ()`
- int `is_choice ()`
- int `is_terminal ()`
- int `is_nonterm ()`
- int `is_concat ()`
- int `is_null ()`
- int `is_loop ()`
- int `is_row ()`
- int `is_production ()`
- int `is_newline ()`
- int `is_rail ()`
- string `east ()`
- string `west ()`
- virtual `coordinate place` (ofstream &outs, int draw, int drawrails, `coordinate` start, `node *parent`, int depth)
- virtual void `fixSkips ()`
- virtual void `insert (node *)`
- virtual void `mergeRails ()`
- virtual void `dump` (int depth) const
- virtual string `texName ()`
- virtual string `rawName ()`
- float `width ()`
- void `setWidth` (float w)
- float `height ()`
- void `setheight` (float h)
- virtual int `mergeConcats` (int depth)
- virtual int `liftConcats` (int depth)
- virtual int `analyzeOptLoops` (int depth)
- virtual int `analyzeNonOptLoops` (int depth)
- virtual int `mergeChoices` (int depth)
- virtual int `numChildren ()`
- virtual `node * getChild` (int n)
- virtual int `operator==` (`node &r`)
- virtual int `operator!=` (`node &r`)
- virtual `node * subsume` (regex_t *name, `node *replacement`)
- virtual void `forgetChild` (int n)
- virtual `node * createRows ()`

Static Public Member Functions

- static void `loadData` (string filename)
- static void `deleteData ()`
- static float `getColSep ()`

Protected Types

- enum `nodetype` {
`GRAMMAR` , `CHOICE` , `TERMINAL` , `NONTERM` ,
`CONCAT` , `NULLNODE` , `LOOP` , `NEWLINE` ,
`PRODUCTION` , `RAIL` , `ROW` , `UNKNOWN` }

Protected Member Functions

- template<class ... Args>
void `line` (ofstream &outs, Args ... args)
- int `same_type` (node &r)

Static Protected Member Functions

- static string `vrailStr` (vraildir d)

Protected Attributes

- `nodetype` type
- string `nodename`
- string `ea`
- string `wa`
- float `myWidth`
- float `myHeight`
- node * `parent`
- node * `previous`
- node * `next`
- float `beforeskip`
- int `drawtoprev`
- railnode * `leftrail`
- railnode * `rightrail`
- int `dead`
- coordinate location

Static Protected Attributes

- static `nodesizes` * `sizes`

9.17.1 Member Enumeration Documentation

9.17.1.1 `nodetype`

enum `node::nodetype` [protected]

Enumerator

GRAMMAR	
CHOICE	
TERMINAL	
NONTERM	
CONCAT	
NULLNODE	
LOOP	
NEWLINE	
PRODUCTION	
RAIL	
ROW	
UNKNOWN	

9.17.2 Constructor & Destructor Documentation

9.17.2.1 node() [1/2]

```
node::node ( )
```

9.17.2.2 node() [2/2]

```
node::node (
    const node & original )
```

9.17.2.3 ~node()

```
node::~~node ( ) [virtual]
```

9.17.3 Member Function Documentation

9.17.3.1 analyzeNonOptLoops()

```
int node::analyzeNonOptLoops (
    int depth ) [virtual]
```

Reimplemented in [concatnode](#), [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.2 analyzeOptLoops()

```
int node::analyzeOptLoops (
    int depth ) [virtual]
```

Reimplemented in [concatnode](#), [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.3 clone()

```
virtual node\* node::clone ( ) const [pure virtual]
```

Implemented in [productionnode](#), [concatnode](#), [loopnode](#), [choicenode](#), [rownode](#), [newlinenode](#), [nullnode](#), [termnode](#), [nontermnode](#), [multinode](#), [railnode](#), and [singlenode](#).

9.17.3.4 createRows()

```
virtual node\* node::createRows ( ) [inline], [virtual]
```

Reimplemented in [productionnode](#), and [concatnode](#).

9.17.3.5 deleteData()

```
static void node::deleteData ( ) [inline], [static]
```

9.17.3.6 drawToLeftRail()

```
void node::drawToLeftRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented in [concatnode](#), [loopnode](#), [choicenode](#), [nontermnode](#), and [singlenode](#).

9.17.3.7 drawToRightRail()

```
void node::drawToRightRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented in [concatnode](#), [loopnode](#), [choicenode](#), [nontermnode](#), and [singlenode](#).

9.17.3.8 dump()

```
void node::dump (
    int depth ) const [virtual]
```

Reimplemented in [productionnode](#), [concatnode](#), [loopnode](#), [choicenode](#), [rownode](#), [newlinenode](#), [nontermnode](#), and [railnode](#).

9.17.3.9 east()

```
string node::east ( ) [inline]
```

9.17.3.10 fixSkips()

```
virtual void node::fixSkips ( ) [inline], [virtual]
```

Reimplemented in [productionnode](#), [concatnode](#), [loopnode](#), [choicenode](#), [multinode](#), and [singlenode](#).

9.17.3.11 forgetChild()

```
virtual void node::forgetChild (
    int n ) [inline], [virtual]
```

Reimplemented in [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.12 getBeforeSkip()

```
float node::getBeforeSkip ( ) [inline]
```

9.17.3.13 getChild()

```
virtual node\* node::getChild (
    int n ) [inline], [virtual]
```

Reimplemented in [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.14 getColSep()

```
static float node::getColSep ( ) [inline], [static]
```

9.17.3.15 getDrawToPrev()

```
int node::getDrawToPrev ( ) [inline]
```

9.17.3.16 getLeftRail()

```
railnode\* node::getLeftRail ( ) [inline]
```

9.17.3.17 getNext()

```
node\* node::getNext ( ) [inline]
```

9.17.3.18 getParent()

```
node\* node::getParent ( ) [inline]
```

9.17.3.19 getPrevious()

```
node\* node::getPrevious ( ) [inline]
```

9.17.3.20 getRightRail()

```
railnode* node::getRightRail ( ) [inline]
```

9.17.3.21 height()

```
float node::height ( ) [inline]
```

9.17.3.22 insert()

```
virtual void node::insert (
    node * ) [inline], [virtual]
```

Reimplemented in [concatnode](#), [choicenode](#), and [multinode](#).

9.17.3.23 is_choice()

```
int node::is_choice ( ) [inline]
```

9.17.3.24 is_concat()

```
int node::is_concat ( ) [inline]
```

9.17.3.25 is_loop()

```
int node::is_loop ( ) [inline]
```

9.17.3.26 is_newline()

```
int node::is_newline ( ) [inline]
```

9.17.3.27 is_nonterm()

```
int node::is_nonterm ( ) [inline]
```

9.17.3.28 is_null()

```
int node::is_null ( ) [inline]
```

9.17.3.29 is_production()

```
int node::is_production ( ) [inline]
```

9.17.3.30 is_rail()

```
int node::is_rail ( ) [inline]
```

9.17.3.31 is_row()

```
int node::is_row ( ) [inline]
```

9.17.3.32 is_terminal()

```
int node::is_terminal ( ) [inline]
```

9.17.3.33 isDead()

```
int node::isDead ( ) [inline]
```

9.17.3.34 liftConcats()

```
int node::liftConcats (
    int depth ) [virtual]
```

Reimplemented in [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.35 line()

```
template<class ... Args>
void node::line (
    ostream & outs,
    Args ... args ) [protected]
```

9.17.3.36 loadData()

```
static void node::loadData (
    string filename ) [inline], [static]
```

9.17.3.37 makeDead()

```
void node::makeDead ( ) [inline]
```

9.17.3.38 mergeChoices()

```
int node::mergeChoices (
    int depth ) [virtual]
```

Reimplemented in [choicenode](#), [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.39 mergeConcats()

```
int node::mergeConcats (
    int depth ) [virtual]
```

Reimplemented in [concatnode](#), [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.40 mergeRails()

```
virtual void node::mergeRails ( ) [inline], [virtual]
```

Reimplemented in [concatnode](#), [multinode](#), and [singlenode](#).

9.17.3.41 numChildren()

```
virtual int node::numChildren ( ) [inline], [virtual]
```

Reimplemented in [multinode](#), and [singlenode](#).

9.17.3.42 operator"!="()

```
virtual int node::operator!= (
    node & r ) [inline], [virtual]
```

Reimplemented in [nontermnode](#), [multinode](#), [railnode](#), and [singlenode](#).

9.17.3.43 operator=="()

```
virtual int node::operator== (
    node & r ) [inline], [virtual]
```

Reimplemented in [nontermnode](#), [multinode](#), [railnode](#), and [singlenode](#).

9.17.3.44 place()

```
virtual coordinate node::place (
    ostream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [inline], [virtual]
```

Reimplemented in [productionnode](#), [concatnode](#), [rownode](#), [newlinenode](#), [nullnode](#), [nontermnode](#), [multinode](#), and [railnode](#).

9.17.3.45 rawName()

```
virtual string node::rawName ( ) [inline], [virtual]
```

9.17.3.46 same_type()

```
int node::same_type (
    node & r ) [inline], [protected]
```

9.17.3.47 setBeforeSkip()

```
void node::setBeforeSkip (
    float s ) [inline]
```

9.17.3.48 setDrawToPrev()

```
void node::setDrawToPrev (
    int d ) [inline]
```

9.17.3.49 setheight()

```
void node::setheight (
    float h ) [inline]
```

9.17.3.50 setLeftRail()

```
virtual void node::setLeftRail (
    railnode * p ) [inline], [virtual]
```

9.17.3.51 setNext()

```
virtual void node::setNext (
    node * p ) [inline], [virtual]
```

Reimplemented in [concatnode](#), [multinode](#), and [singlenode](#).

9.17.3.52 setParent()

```
virtual void node::setParent (
    node * p ) [inline], [virtual]
```

Reimplemented in [multinode](#), and [singlenode](#).

9.17.3.53 setPrevious()

```
virtual void node::setPrevious (
    node * p ) [inline], [virtual]
```

Reimplemented in [concatnode](#), [multinode](#), and [singlenode](#).

9.17.3.54 setRightRail()

```
virtual void node::setRightRail (
    railnode * p ) [inline], [virtual]
```

9.17.3.55 setwidth()

```
void node::setwidth (
    float w ) [inline]
```

9.17.3.56 subsume()

```
virtual node* node::subsume (
    regex_t * name,
    node * replacement ) [inline], [virtual]
```

Reimplemented in [productionnode](#), [nontermnode](#), [multinode](#), and [singlenode](#).

9.17.3.57 texName()

```
virtual string node::texName ( ) [inline], [virtual]
```

Reimplemented in [productionnode](#), [loopnode](#), [choicenode](#), [rownode](#), [nullnode](#), [nontermnode](#), [multinode](#), [railnode](#), and [singlenode](#).

9.17.3.58 vrailStr()

```
static string node::vrailStr (
    vraildir d ) [inline], [static], [protected]
```

9.17.3.59 west()

```
string node::west ( ) [inline]
```

9.17.3.60 width()

```
float node::width ( ) [inline]
```

9.17.4 Member Data Documentation

9.17.4.1 beforeskip

```
float node::beforeskip [protected]
```

9.17.4.2 dead

```
int node::dead [protected]
```

9.17.4.3 drawtoprev

```
int node::drawtoprev [protected]
```

9.17.4.4 ea

```
string node::ea [protected]
```

9.17.4.5 leftrail

`railnode*` `node::leftrail` [protected]

9.17.4.6 location

`coordinate` `node::location` [protected]

9.17.4.7 myHeight

`float` `node::myHeight` [protected]

9.17.4.8 myWidth

`float` `node::myWidth` [protected]

9.17.4.9 next

`node*` `node::next` [protected]

9.17.4.10 nodename

`string` `node::nodename` [protected]

9.17.4.11 parent

`node*` `node::parent` [protected]

9.17.4.12 previous

`node*` `node::previous` [protected]

9.17.4.13 rightrail

`railnode * node::rightrail` [protected]

9.17.4.14 sizes

`nodesizes * node::sizes` [static], [protected]

9.17.4.15 type

`nodetype node::type` [protected]

9.17.4.16 wa

`string node::wa` [protected]

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

- [graph.hh](#)
- [graph.cc](#)
- [optimize.cc](#)
- [output.cc](#)

9.18 nodesizes Class Reference

```
#include <nodesize.hh>
```

Public Member Functions

- [nodesizes](#) ()
- [~nodesizes](#) ()
- void [loadData](#) (string filename)
- int [getSize](#) (string nodename, float &width, float &height)

Public Attributes

- float [rowsep](#)
- float [colsep](#)
- float [minsize](#)

9.18.1 Constructor & Destructor Documentation

9.18.1.1 nodesizes()

```
nodesizes::nodesizes ( ) [inline]
```

9.18.1.2 ~nodesizes()

```
nodesizes::~~nodesizes ( ) [inline]
```

9.18.2 Member Function Documentation

9.18.2.1 getSize()

```
int nodesizes::getSize (
    string nodename,
    float & width,
    float & height ) [inline]
```

9.18.2.2 loadData()

```
void nodesizes::loadData (
    string filename ) [inline]
```

9.18.3 Member Data Documentation

9.18.3.1 colsep

```
float nodesizes::colsep
```

9.18.3.2 minsize

```
float nodesizes::minsize
```

9.18.3.3 rowsep

```
float nodesizes::rowsep
```

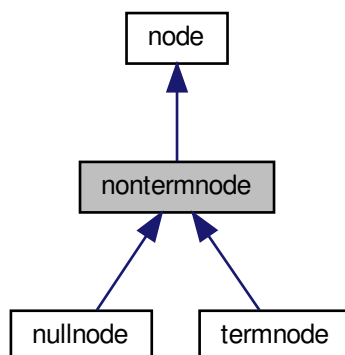
The documentation for this class was generated from the following file:

- [nodesize.hh](#)

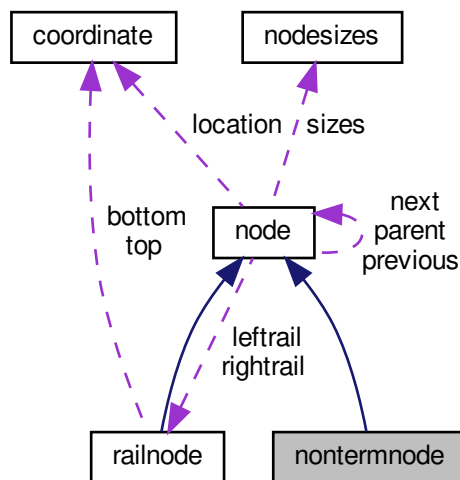
9.19 nontermnode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for nontermnode:



Collaboration diagram for nontermnode:



Public Member Functions

- `nontermnode` (string s)
- `nontermnode` (const `nontermnode` &original)
- virtual `nontermnode * clone` () const
- virtual `~nontermnode` ()
- virtual void `forgetChild` (int n)
- virtual void `dump` (int depth) const
- virtual string `texName` ()
- virtual `coordinate place` (ofstream &outs, int draw, int drawrails, `coordinate` start, `node *parent`, int depth)
- virtual int `mergeConcats` (int depth)
- virtual int `mergeChoices` (int depth)
- virtual int `liftConcats` (int depth)
- virtual `node * getChild` (int n)
- virtual int `analyzeOptLoops` (int depth)
- virtual int `analyzeNonOptLoops` (int depth)
- virtual void `drawToLeftRail` (ofstream &outs, `railnode *p`, `vraildir` join, int drawself)
- virtual void `drawToRightRail` (ofstream &outs, `railnode *p`, `vraildir` join, int drawself)
- virtual int `operator==` (`node` &r)
- virtual int `operator!=` (`node` &r)
- virtual `node * subsume` (regex_t *name, `node` *replacement)

Protected Attributes

- string `style`
- string `format`
- string `str`

Additional Inherited Members

9.19.1 Constructor & Destructor Documentation

9.19.1.1 nontermnode() [1/2]

```
nontermnode::nontermnode (
    string s )
```

9.19.1.2 nontermnode() [2/2]

```
nontermnode::nontermnode (
    const nontermnode & original )
```

9.19.1.3 ~nontermnode()

```
virtual nontermnode::~~nontermnode ( ) [inline], [virtual]
```

9.19.2 Member Function Documentation

9.19.2.1 analyzeNonOptLoops()

```
virtual int nontermnode::analyzeNonOptLoops (
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.2 analyzeOptLoops()

```
virtual int nontermnode::analyzeOptLoops (
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.3 clone()

```
nontermnode * nontermnode::clone ( ) const [virtual]
```

Implements [node](#).

Reimplemented in [nullnode](#), and [termnode](#).

9.19.2.4 drawToLeftRail()

```
void nontermnode::drawToLeftRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.19.2.5 drawToRightRail()

```
void nontermnode::drawToRightRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.19.2.6 dump()

```
void nontermnode::dump (
    int depth ) const [virtual]
```

Reimplemented from [node](#).

9.19.2.7 forgetChild()

```
virtual void nontermnode::forgetChild (
    int n ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.8 getChild()

```
virtual node* nontermnode::getChild (
    int n ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.9 liftConcats()

```
virtual int nontermnode::liftConcats (
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.10 mergeChoices()

```
virtual int nontermnode::mergeChoices (
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.11 mergeConcats()

```
virtual int nontermnode::mergeConcats (
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.12 operator"!="()

```
virtual int nontermnode::operator!= (
    node & r ) [inline], [virtual]
```

Reimplemented from [node](#).

9.19.2.13 operator==()

```
int nontermnode::operator== (
    node & r ) [virtual]
```

Reimplemented from [node](#).

9.19.2.14 place()

```
coordinate nontermnode::place (
    ostream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [nullnode](#).

9.19.2.15 subsume()

```
node * nontermnode::subsume (
    regex_t * name,
    node * replacement ) [virtual]
```

Reimplemented from [node](#).

9.19.2.16 texName()

```
virtual string nontermnode::texName ( ) [inline], [virtual]
```

Reimplemented from [node](#).

Reimplemented in [nullnode](#).

9.19.3 Member Data Documentation

9.19.3.1 format

```
string nontermnode::format [protected]
```

9.19.3.2 str

```
string nontermnode::str [protected]
```

9.19.3.3 style

```
string nontermnode::style [protected]
```

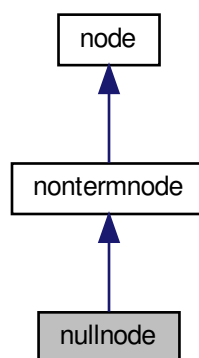
The documentation for this class was generated from the following files:

- [graph.hh](#)
- [graph.cc](#)
- [output.cc](#)
- [subsume.cc](#)

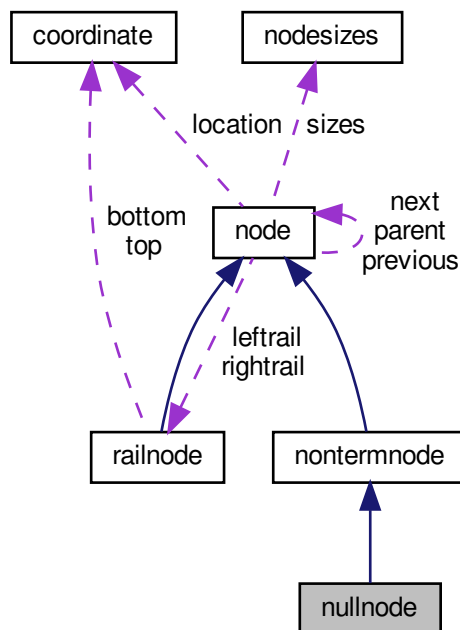
9.20 nullnode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for nullnode:



Collaboration diagram for nullnode:



Public Member Functions

- [nullnode](#) (string s)
- [nullnode](#) (const [nullnode](#) &original)
- virtual [nullnode](#) * [clone](#) () const
- virtual [coordinate place](#) (ofstream &outs, int draw, int drawrails, [coordinate](#) start, [node](#) *parent, int depth)
- virtual string [texName](#) ()

Additional Inherited Members

9.20.1 Constructor & Destructor Documentation

9.20.1.1 nullnode() [1/2]

```

nullnode::nullnode (
    string s )

```

9.20.1.2 nullnode() [2/2]

```
nullnode::nullnode (
    const nullnode & original )
```

9.20.2 Member Function Documentation

9.20.2.1 clone()

```
nullnode * nullnode::clone ( ) const [virtual]
```

Reimplemented from [nontermnode](#).

9.20.2.2 place()

```
coordinate nullnode::place (
    ostream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [nontermnode](#).

9.20.2.3 texName()

```
virtual string nullnode::texName ( ) [inline], [virtual]
```

Reimplemented from [nontermnode](#).

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

- [graph.hh](#)
- [graph.cc](#)
- [output.cc](#)

9.21 annot::parser Class Reference

A Bison parser.

```
#include <annot_parser.hh>
```

Classes

- struct [basic_symbol](#)
- struct [by_kind](#)
Type access provider for token (enum) based symbols.
- class [context](#)
- class [semantic_type](#)
- struct [symbol_kind](#)
Symbol kinds.
- struct [symbol_type](#)
"External" symbols: returned by the scanner.
- struct [syntax_error](#)
Syntax errors thrown from user actions.
- struct [token](#)
Token kinds.

Public Types

- typedef [location](#) [location_type](#)
Symbol locations.
- typedef [token::yytokentype](#) [token_kind_type](#)
Token kind, as returned by yylex.
- typedef [token_kind_type](#) [token_type](#)
Backward compatibility alias (Bison 3.6).
- typedef [symbol_kind::symbol_kind_type](#) [symbol_kind_type](#)
(Internal) symbol kind.
- typedef [by_kind](#) [by_type](#)
Backward compatibility for a private implementation detail (Bison 3.6).

Public Member Functions

- [parser](#) ([annotmap](#) *m_yyarg)
Build a parser object.
- virtual [~parser](#) ()
- int [operator\(\)](#) ()
- virtual int [parse](#) ()
- virtual void [error](#) (const [location_type](#) &loc, const std::string &msg)
- void [error](#) (const [syntax_error](#) &err)
Report a syntax error.

Static Public Member Functions

- static const char * [symbol_name](#) ([symbol_kind_type](#) yysymbol)
- static [symbol_type](#) [make_END](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_ANNOTerror](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_ANNOTUNDEF](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_ASTART](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_AEND](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_SEMICOLON](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_SUBSUME](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_AS](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_CAPTION](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_SIDEWAYS](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_UNEXP](#) (const std::string &v, const [location_type](#) &l)
- static [symbol_type](#) [make_STRING](#) (const std::string &v, const [location_type](#) &l)

Static Public Attributes

- static const [symbol_kind_type](#) YYNTOKENS = symbol_kind::YYNTOKENS
The number of tokens.

9.21.1 Detailed Description

A Bison parser.

9.21.2 Member Typedef Documentation

9.21.2.1 by_type

```
typedef by\_kind annot::parser::by\_type
```

Backward compatibility for a private implementation detail (Bison 3.6).

9.21.2.2 location_type

```
typedef location annot::parser::location\_type
```

Symbol locations.

9.21.2.3 symbol_kind_type

```
typedef symbol\_kind::symbol\_kind\_type annot::parser::symbol\_kind\_type
```

(Internal) symbol kind.

9.21.2.4 token_kind_type

```
typedef token::yytokentype annot::parser::token\_kind\_type
```

Token kind, as returned by yylex.

9.21.2.5 token_type

```
typedef token_kind_type annot::parser::token_type
```

Backward compatibility alias (Bison 3.6).

9.21.3 Constructor & Destructor Documentation

9.21.3.1 parser()

```
annot::parser::parser (
    annotmap * m_yyarg )
```

Build a parser object.

9.21.3.2 ~parser()

```
annot::parser::~parser ( ) [virtual]
```

9.21.4 Member Function Documentation

9.21.4.1 error() [1/2]

```
void annot::parser::error (
    const location_type & loc,
    const std::string & msg ) [virtual]
```

Report a syntax error.

Parameters

<i>loc</i>	where the syntax error is found.
<i>msg</i>	a description of the syntax error.

9.21.4.2 error() [2/2]

```
void annot::parser::error (
    const syntax_error & err )
```


Report a syntax error.

9.21.4.3 make_AEND()

```
static symbol_type annot::parser::make_AEND (
    const location_type & l ) [inline], [static]
```

9.21.4.4 make_ANNOTerror()

```
static symbol_type annot::parser::make_ANNOTerror (
    const location_type & l ) [inline], [static]
```

9.21.4.5 make_ANNOTUNDEF()

```
static symbol_type annot::parser::make_ANNOTUNDEF (
    const location_type & l ) [inline], [static]
```

9.21.4.6 make_AS()

```
static symbol_type annot::parser::make_AS (
    const location_type & l ) [inline], [static]
```

9.21.4.7 make_ASTART()

```
static symbol_type annot::parser::make_ASTART (
    const location_type & l ) [inline], [static]
```

9.21.4.8 make_CAPTION()

```
static symbol_type annot::parser::make_CAPTION (
    const location_type & l ) [inline], [static]
```

9.21.4.9 make_END()

```
static symbol_type annot::parser::make_END (
    const location_type & l ) [inline], [static]
```

9.21.4.10 make_SEMICOLON()

```
static symbol_type annot::parser::make_SEMICOLON (
    const location_type & l ) [inline], [static]
```

9.21.4.11 make_SIDEWAYS()

```
static symbol_type annot::parser::make_SIDEWAYS (
    const location_type & l ) [inline], [static]
```

9.21.4.12 make_STRING()

```
static symbol_type annot::parser::make_STRING (
    const std::string & v,
    const location_type & l ) [inline], [static]
```

9.21.4.13 make_SUBSUME()

```
static symbol_type annot::parser::make_SUBSUME (
    const location_type & l ) [inline], [static]
```

9.21.4.14 make_UNEXP()

```
static symbol_type annot::parser::make_UNEXP (
    const std::string & v,
    const location_type & l ) [inline], [static]
```

9.21.4.15 operator>()

```
int annot::parser::operator() ( )
```

Parse. An alias for parse ().

Returns

0 iff parsing succeeded.

9.21.4.16 parse()

```
int annot::parser::parse ( ) [virtual]
```

Parse.

Returns

0 iff parsing succeeded.

Length of the RHS of the rule being reduced.

The lookahead symbol.

The locations where the error started and ended.

The return value of parse ().

Discard the LAC context in case there still is one left from a previous invocation.

9.21.4.17 symbol_name()

```
const char * annot::parser::symbol_name (
    symbol_kind_type yysymbol ) [static]
```

The user-facing name of the symbol whose (internal) number is YYSYMBOL. No bounds checking.

9.21.5 Member Data Documentation

9.21.5.1 YYNTOKENS

```
const symbol_kind_type annot::parser::YYNTOKENS = symbol_kind::YYNTOKENS [static]
```

The number of tokens.

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

- [annot_parser.hh](#)
- [annot_parser.cc](#)

9.22 yy::parser Class Reference

A Bison parser.

```
#include <parser.hh>
```

Classes

- struct [basic_symbol](#)
- struct [by_kind](#)
Type access provider for token (enum) based symbols.
- class [context](#)
- struct [symbol_kind](#)
Symbol kinds.
- struct [symbol_type](#)
"External" symbols: returned by the scanner.
- struct [syntax_error](#)
Syntax errors thrown from user actions.
- struct [token](#)
Token kinds.
- class [value_type](#)

Public Types

- typedef [value_type](#) [semantic_type](#)
Backward compatibility (Bison 3.8).
- typedef [location](#) [location_type](#)
Symbol locations.
- typedef [token::token_kind_type](#) [token_kind_type](#)
Token kind, as returned by yylex.
- typedef [token_kind_type](#) [token_type](#)
Backward compatibility alias (Bison 3.6).
- typedef [symbol_kind::symbol_kind_type](#) [symbol_kind_type](#)
(Internal) symbol kind.
- typedef [by_kind](#) [by_type](#)
Backward compatibility for a private implementation detail (Bison 3.6).
- typedef int [debug_level_type](#)
Type for debugging levels.

Public Member Functions

- [parser](#) ([driver](#) &drv_yyarg)
Build a parser object.
- virtual [~parser](#) ()
- int [operator\(\)](#) ()
- virtual int [parse](#) ()
- std::ostream & [debug_stream](#) () const [YY_ATTRIBUTE_PURE](#)
The current debugging stream.
- void [set_debug_stream](#) (std::ostream &)
Set the current debugging stream.
- [debug_level_type](#) [debug_level](#) () const [YY_ATTRIBUTE_PURE](#)
The current debugging level.
- void [set_debug_level](#) ([debug_level_type](#) l)
Set the current debugging level.
- virtual void [error](#) (const [location_type](#) &loc, const std::string &msg)
- void [error](#) (const [syntax_error](#) &err)
Report a syntax error.

Static Public Member Functions

- static const char * [symbol_name](#) ([symbol_kind_type](#) yysymbol)
- static [symbol_type](#) [make_END](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_YYerror](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_YYUNDEF](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_COMMA](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_EQUAL](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_SEMICOLON](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_PIPE](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_LBRACK](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_RBRACK](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_LPAREN](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_RPAREN](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_LBRACE](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_RBRACE](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_NEWLINE](#) (const [location_type](#) &l)
- static [symbol_type](#) [make_UNEXP](#) (const std::string &v, const [location_type](#) &l)
- static [symbol_type](#) [make_TERM](#) (const std::string &v, const [location_type](#) &l)
- static [symbol_type](#) [make_STRING](#) (const std::string &v, const [location_type](#) &l)
- static [symbol_type](#) [make_ANNOTATION](#) (const std::string &v, const [location_type](#) &l)

Static Public Attributes

- static const [symbol_kind_type](#) [YYNTOKENS](#) = [symbol_kind::YYNTOKENS](#)
The number of tokens.

9.22.1 Detailed Description

A Bison parser.

9.22.2 Member Typedef Documentation

9.22.2.1 by_type

```
typedef by_kind yy::parser::by_type
```

Backward compatibility for a private implementation detail (Bison 3.6).

9.22.2.2 debug_level_type

```
typedef int yy::parser::debug_level_type
```

Type for debugging levels.

9.22.2.3 location_type

```
typedef location yy::parser::location_type
```

Symbol locations.

9.22.2.4 semantic_type

```
typedef value_type yy::parser::semantic_type
```

Backward compatibility (Bison 3.8).

9.22.2.5 symbol_kind_type

```
typedef symbol_kind::symbol_kind_type yy::parser::symbol_kind_type
```

(Internal) symbol kind.

9.22.2.6 token_kind_type

```
typedef token::token_kind_type yy::parser::token_kind_type
```

Token kind, as returned by yylex.

9.22.2.7 token_type

```
typedef token_kind_type yy::parser::token_type
```

Backward compatibility alias (Bison 3.6).

9.22.3 Constructor & Destructor Documentation

9.22.3.1 parser()

```
yy::parser::parser (
    driver & drv_yyarg )
```

Build a parser object.

9.22.3.2 ~parser()

```
yy::parser::~~parser ( ) [virtual]
```

9.22.4 Member Function Documentation

9.22.4.1 debug_level()

```
parser::debug_level_type yy::parser::debug_level ( ) const
```

The current debugging level.

9.22.4.2 debug_stream()

```
std::ostream & yy::parser::debug_stream ( ) const
```

The current debugging stream.

9.22.4.3 error() [1/2]

```
void yy::parser::error (
    const location_type & loc,
    const std::string & msg ) [virtual]
```

Report a syntax error.

Parameters

<i>loc</i>	where the syntax error is found.
<i>msg</i>	a description of the syntax error.

9.22.4.4 error() [2/2]

```
void yy::parser::error (
    const syntax\_error & err )
```

Report a syntax error.

9.22.4.5 make_ANNOTATION()

```
static symbol\_type yy::parser::make_ANNOTATION (
    const std::string & v,
    const location\_type & l ) [inline], [static]
```

9.22.4.6 make_COMMA()

```
static symbol\_type yy::parser::make_COMMA (
    const location\_type & l ) [inline], [static]
```

9.22.4.7 make_END()

```
static symbol\_type yy::parser::make_END (
    const location\_type & l ) [inline], [static]
```

9.22.4.8 make_EQUAL()

```
static symbol\_type yy::parser::make_EQUAL (
    const location\_type & l ) [inline], [static]
```


9.22.4.9 make_LBRACE()

```
static symbol_type yy::parser::make_LBRACE (  
    const location_type & l ) [inline], [static]
```

9.22.4.10 make_LBRACK()

```
static symbol_type yy::parser::make_LBRACK (  
    const location_type & l ) [inline], [static]
```

9.22.4.11 make_LPAREN()

```
static symbol_type yy::parser::make_LPAREN (  
    const location_type & l ) [inline], [static]
```

9.22.4.12 make_NEWLINE()

```
static symbol_type yy::parser::make_NEWLINE (  
    const location_type & l ) [inline], [static]
```

9.22.4.13 make_PIPE()

```
static symbol_type yy::parser::make_PIPE (  
    const location_type & l ) [inline], [static]
```

9.22.4.14 make_RBRACE()

```
static symbol_type yy::parser::make_RBRACE (  
    const location_type & l ) [inline], [static]
```

9.22.4.15 make_RBRACK()

```
static symbol_type yy::parser::make_RBRACK (  
    const location_type & l ) [inline], [static]
```

9.22.4.16 make_RPAREN()

```
static symbol_type yy::parser::make_RPAREN (  
    const location_type & l ) [inline], [static]
```

9.22.4.17 make_SEMICOLON()

```
static symbol_type yy::parser::make_SEMICOLON (  
    const location_type & l ) [inline], [static]
```

9.22.4.18 make_STRING()

```
static symbol_type yy::parser::make_STRING (  
    const std::string & v,  
    const location_type & l ) [inline], [static]
```

9.22.4.19 make_TERM()

```
static symbol_type yy::parser::make_TERM (  
    const std::string & v,  
    const location_type & l ) [inline], [static]
```

9.22.4.20 make_UNEXP()

```
static symbol_type yy::parser::make_UNEXP (  
    const std::string & v,  
    const location_type & l ) [inline], [static]
```

9.22.4.21 make_YYerror()

```
static symbol_type yy::parser::make_YYerror (  
    const location_type & l ) [inline], [static]
```

9.22.4.22 make_YYUNDEF()

```
static symbol_type yy::parser::make_YYUNDEF (
    const location_type & l ) [inline], [static]
```

9.22.4.23 operator>()

```
int yy::parser::operator() ( )
```

Parse. An alias for parse ().

Returns

0 iff parsing succeeded.

9.22.4.24 parse()

```
int yy::parser::parse ( ) [virtual]
```

Parse.

Returns

0 iff parsing succeeded.

Length of the RHS of the rule being reduced.

The lookahead symbol.

The locations where the error started and ended.

The return value of parse ().

9.22.4.25 set_debug_level()

```
void yy::parser::set_debug_level (
    debug_level_type l )
```

Set the current debugging level.

9.22.4.26 `set_debug_stream()`

```
void yy::parser::set_debug_stream (
    std::ostream & o )
```

Set the current debugging stream.

9.22.4.27 `symbol_name()`

```
const char * yy::parser::symbol_name (
    symbol_kind_type yysymbol ) [static]
```

The user-facing name of the symbol whose (internal) number is YYSYMBOL. No bounds checking.

9.22.5 Member Data Documentation

9.22.5.1 YYNTOKENS

```
const symbol_kind_type yy::parser::YYNTOKENS = symbol_kind::YYNTOKENS [static]
```

The number of tokens.

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

- [parser.hh](#)
- [parser.cc](#)

9.23 `annot::position` Class Reference

A point in a source file.

```
#include <annot_location.hh>
```

Public Types

- typedef const std::string [filename_type](#)
Type for file name.
- typedef int [counter_type](#)
Type for line and column numbers.

Public Member Functions

- `position (filename_type *f=YY_NULLPTR, counter_type l=1, counter_type c=1)`
Construct a position.
- `void initialize (filename_type *fn=YY_NULLPTR, counter_type l=1, counter_type c=1)`
Initialization.

Line and Column related manipulators

- `void lines (counter_type count=1)`
(line related) Advance to the COUNT next lines.
- `void columns (counter_type count=1)`
(column related) Advance to the COUNT next columns.

Public Attributes

- `filename_type * filename`
File name to which this position refers.
- `counter_type line`
Current line number.
- `counter_type column`
Current column number.

9.23.1 Detailed Description

A point in a source file.

9.23.2 Member Typedef Documentation

9.23.2.1 counter_type

```
typedef int annot::position::counter_type
```

Type for line and column numbers.

9.23.2.2 filename_type

```
typedef const std::string annot::position::filename_type
```

Type for file name.

9.23.3 Constructor & Destructor Documentation

9.23.3.1 position()

```
annot::position::position (
    filename_type * f = YY_NULLPTR,
    counter_type l = 1,
    counter_type c = 1 ) [inline], [explicit]
```

Construct a position.

9.23.4 Member Function Documentation

9.23.4.1 columns()

```
void annot::position::columns (
    counter_type count = 1 ) [inline]
```

(column related) Advance to the COUNT next columns.

9.23.4.2 initialize()

```
void annot::position::initialize (
    filename_type * fn = YY_NULLPTR,
    counter_type l = 1,
    counter_type c = 1 ) [inline]
```

Initialization.

9.23.4.3 lines()

```
void annot::position::lines (
    counter_type count = 1 ) [inline]
```

(line related) Advance to the COUNT next lines.

9.23.5 Member Data Documentation

9.23.5.1 column

`counter_type` `annot::position::column`

Current column number.

9.23.5.2 filename

`filename_type*` `annot::position::filename`

File name to which this position refers.

9.23.5.3 line

`counter_type` `annot::position::line`

Current line number.

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

- [annot_location.hh](#)

9.24 yy::position Class Reference

A point in a source file.

```
#include <location.hh>
```

Public Types

- typedef const std::string `filename_type`
Type for file name.
- typedef int `counter_type`
Type for line and column numbers.

Public Member Functions

- `position` (`filename_type` *f=YY_NULLPTR, `counter_type` l=1, `counter_type` c=1)
Construct a position.
- void `initialize` (`filename_type` *fn=YY_NULLPTR, `counter_type` l=1, `counter_type` c=1)
Initialization.

Line and Column related manipulators

- void `lines` (`counter_type` count=1)
(line related) Advance to the COUNT next lines.
- void `columns` (`counter_type` count=1)
(column related) Advance to the COUNT next columns.

Public Attributes

- [filename_type](#) * [filename](#)
File name to which this position refers.
- [counter_type](#) [line](#)
Current line number.
- [counter_type](#) [column](#)
Current column number.

9.24.1 Detailed Description

A point in a source file.

9.24.2 Member Typedef Documentation

9.24.2.1 counter_type

```
typedef int yy::position::counter_type
```

Type for line and column numbers.

9.24.2.2 filename_type

```
typedef const std::string yy::position::filename_type
```

Type for file name.

9.24.3 Constructor & Destructor Documentation

9.24.3.1 position()

```
yy::position::position (  
    filename_type * f = YY_NULLPTR,  
    counter_type l = 1,  
    counter_type c = 1 ) [inline], [explicit]
```

Construct a position.

9.24.4 Member Function Documentation

9.24.4.1 columns()

```
void yy::position::columns (
    counter_type count = 1 ) [inline]
```

(column related) Advance to the COUNT next columns.

9.24.4.2 initialize()

```
void yy::position::initialize (
    filename_type * fn = YY_NULLPTR,
    counter_type l = 1,
    counter_type c = 1 ) [inline]
```

Initialization.

9.24.4.3 lines()

```
void yy::position::lines (
    counter_type count = 1 ) [inline]
```

(line related) Advance to the COUNT next lines.

9.24.5 Member Data Documentation

9.24.5.1 column

```
counter_type yy::position::column
```

Current column number.

9.24.5.2 filename

```
filename_type* yy::position::filename
```

File name to which this position refers.

9.24.5.3 line

```
counter_type yy::position::line
```

Current line number.

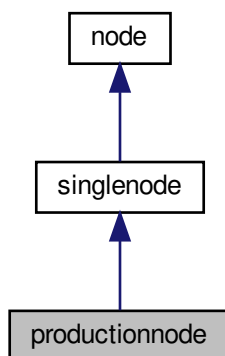
The documentation for this class was generated from the following file:

- [location.hh](#)

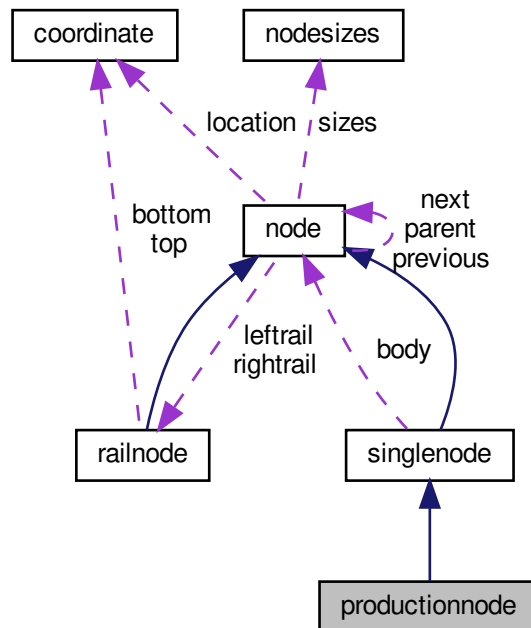
9.25 productionnode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for productionnode:



Collaboration diagram for productionnode:



Public Member Functions

- [productionnode](#) ([annotmap](#) *subsumespec, string s, [node](#) *p)
- [productionnode](#) (const [productionnode](#) &original)
- virtual [productionnode](#) * [clone](#) () const
- virtual [~productionnode](#) ()
- virtual [regex_t](#) * [getSubsume](#) ()
- virtual string [getName](#) ()
- void [optimize](#) ()
- virtual [node](#) * [subsume](#) ([regex_t](#) *name, [node](#) *replacement)
- virtual void [dump](#) (int depth) const
- virtual [coordinate](#) [place](#) (ofstream &outs, int draw, int drawrails, [coordinate](#) start, [node](#) *parent, int depth)
- virtual void [fixSkips](#) ()
- virtual [node](#) * [createRows](#) ()
- virtual string [texName](#) ()

Additional Inherited Members

9.25.1 Constructor & Destructor Documentation

9.25.1.1 productionnode() [1/2]

```
productionnode::productionnode (
    annotmap * subsumespec,
    string s,
    node * p )
```

9.25.1.2 productionnode() [2/2]

```
productionnode::productionnode (
    const productionnode & original )
```

9.25.1.3 ~productionnode()

```
virtual productionnode::~~productionnode ( ) [inline], [virtual]
```

9.25.2 Member Function Documentation

9.25.2.1 clone()

```
productionnode * productionnode::clone ( ) const [virtual]
```

Reimplemented from [singlenode](#).

9.25.2.2 createRows()

```
node * productionnode::createRows ( ) [virtual]
```

Reimplemented from [node](#).

9.25.2.3 dump()

```
void productionnode::dump (
    int depth ) const [virtual]
```

Reimplemented from [node](#).

9.25.2.4 fixSkips()

```
virtual void productionnode::fixSkips ( ) [inline], [virtual]
```

Reimplemented from [singlenode](#).

9.25.2.5 getName()

```
virtual string productionnode::getName ( ) [inline], [virtual]
```

9.25.2.6 getSubsume()

```
virtual regex_t* productionnode::getSubsume ( ) [inline], [virtual]
```

9.25.2.7 optimize()

```
void productionnode::optimize ( )
```

9.25.2.8 place()

```
coordinate productionnode::place (
    ostream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [node](#).

9.25.2.9 subsume()

```
node * productionnode::subsume (
    regex_t * name,
    node * replacement ) [virtual]
```

Reimplemented from [singlenode](#).

9.25.2.10 texName()

```
virtual string productionnode::texName ( ) [inline], [virtual]
```

Reimplemented from [singlenode](#).

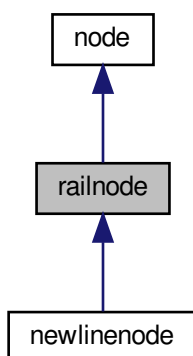
The documentation for this class was generated from the following files:

- [graph.hh](#)
- [graph.cc](#)
- [optimize.cc](#)
- [output.cc](#)
- [subsume.cc](#)

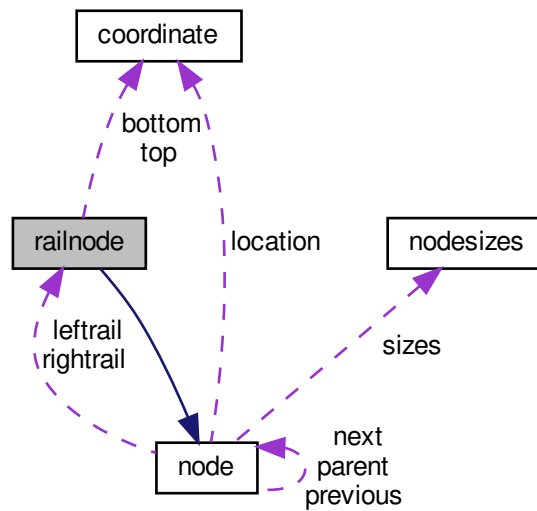
9.26 railnode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for railnode:



Collaboration diagram for railnode:



Public Member Functions

- [railnode](#) ()
- [railnode](#) ([vraillside](#) s, [vrailldir](#) d)
- [railnode](#) (const [railnode](#) &original)
- virtual [railnode](#) * [clone](#) () const
- virtual [~railnode](#) ()
- virtual void [setBottom](#) ([coordinate](#) b)
- virtual [coordinate](#) [getBottom](#) ()
- virtual void [dump](#) (int depth) const
- virtual [coordinate](#) [place](#) (ofstream &outs, int draw, int drawrails, [coordinate](#) start, [node](#) *parent, int depth)
- virtual int [operator==](#) ([node](#) &r)
- virtual int [operator!=](#) ([node](#) &r)
- virtual [vrailldir](#) [getRailDir](#) ()
- virtual void [setRailDir](#) ([vrailldir](#) d)
- virtual string [texName](#) ()

Protected Attributes

- [vraillside](#) side
- [vrailldir](#) direction
- [coordinate](#) top
- [coordinate](#) bottom

Additional Inherited Members

9.26.1 Constructor & Destructor Documentation

9.26.1.1 railnode() [1/3]

```
railnode::railnode ( )
```

9.26.1.2 railnode() [2/3]

```
railnode::railnode (
    vrailside s,
    vraildir d )
```

9.26.1.3 railnode() [3/3]

```
railnode::railnode (
    const railnode & original )
```

9.26.1.4 ~railnode()

```
virtual railnode::~~railnode ( ) [inline], [virtual]
```

9.26.2 Member Function Documentation

9.26.2.1 clone()

```
virtual railnode* railnode::clone ( ) const [inline], [virtual]
```

Implements [node](#).

Reimplemented in [newlinenode](#).

9.26.2.2 dump()

```
void railnode::dump (
    int depth ) const [virtual]
```

Reimplemented from [node](#).

Reimplemented in [newlinenode](#).

9.26.2.3 getBottom()

```
virtual coordinate railnode::getBottom ( ) [inline], [virtual]
```

9.26.2.4 getRailDir()

```
virtual vraildir railnode::getRailDir ( ) [inline], [virtual]
```

9.26.2.5 operator!=(())

```
virtual int railnode::operator!= (
    node & r ) [inline], [virtual]
```

Reimplemented from [node](#).

9.26.2.6 operator==(())

```
int railnode::operator==(
    node & r ) [virtual]
```

Reimplemented from [node](#).

9.26.2.7 place()

```
coordinate railnode::place (
    ofstream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [newlinenode](#).

9.26.2.8 setBottom()

```
virtual void railnode::setBottom (
    coordinate b ) [inline], [virtual]
```

9.26.2.9 setRailDir()

```
virtual void railnode::setRailDir (
    vraildir d ) [inline], [virtual]
```

9.26.2.10 texName()

```
virtual string railnode::texName ( ) [inline], [virtual]
```

Reimplemented from [node](#).

9.26.3 Member Data Documentation

9.26.3.1 bottom

```
coordinate railnode::bottom [protected]
```

9.26.3.2 direction

```
vraildir railnode::direction [protected]
```

9.26.3.3 side

```
vrailside railnode::side [protected]
```

9.26.3.4 top

`coordinate` `railnode::top` [protected]

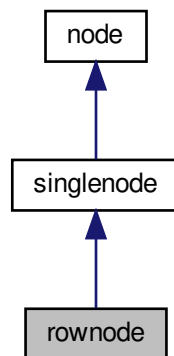
The documentation for this class was generated from the following files:

- [graph.hh](#)
- [graph.cc](#)
- [output.cc](#)

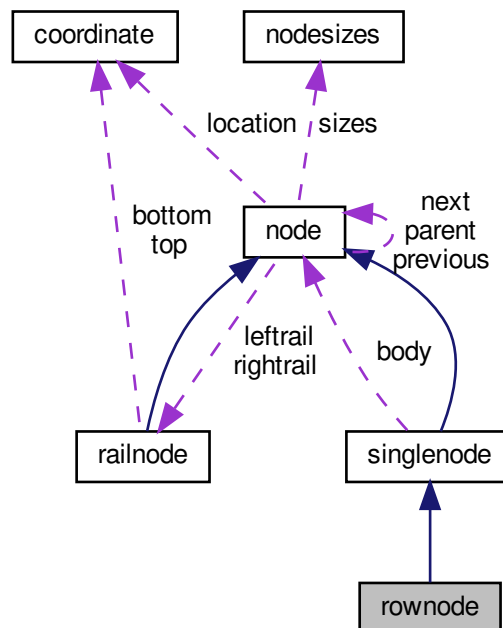
9.27 rownode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for rownode:



Collaboration diagram for rownode:



Public Member Functions

- `rownode (node *p)`
- `rownode (const rownode &original)`
- `virtual rownode * clone () const`
- `virtual ~rownode ()`
- `virtual void dump (int depth) const`
- `virtual coordinate place (ofstream &outs, int draw, int drawrails, coordinate start, node *parent, int depth)`
- `virtual string texName ()`

Additional Inherited Members

9.27.1 Constructor & Destructor Documentation

9.27.1.1 rownode() [1/2]

```

rownode::rownode (
    node * p )

```

9.27.1.2 rownode() [2/2]

```
rownode::rownode (
    const rownode & original )
```

9.27.1.3 ~rownode()

```
virtual rownode::~~rownode ( ) [inline], [virtual]
```

9.27.2 Member Function Documentation

9.27.2.1 clone()

```
rownode * rownode::clone ( ) const [virtual]
```

Reimplemented from [singlenode](#).

9.27.2.2 dump()

```
void rownode::dump (
    int depth ) const [virtual]
```

Reimplemented from [node](#).

9.27.2.3 place()

```
coordinate rownode::place (
    ostream & outs,
    int draw,
    int drawrails,
    coordinate start,
    node * parent,
    int depth ) [virtual]
```

Reimplemented from [node](#).

9.27.2.4 texName()

```
virtual string rownode::texName ( ) [inline], [virtual]
```

Reimplemented from [singlenode](#).

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

- [graph.hh](#)
- [graph.cc](#)
- [output.cc](#)

9.28 annot::parser::semantic_type Class Reference

```
#include <annot_parser.hh>
```

Public Types

- typedef [semantic_type](#) [self_type](#)
*Type of *this.*

Public Member Functions

- [semantic_type](#) () [YY_NOEXCEPT](#)
Empty construction.
- template<typename T >
[semantic_type](#) ([YY_RVREF](#)(T) t)
Construct and fill.
- ~[semantic_type](#) () [YY_NOEXCEPT](#)
Destruction, allowed only if empty.
- template<typename T >
T & [emplace](#) ()
Instantiate an empty T in here.
- template<typename T >
T & [emplace](#) (const T &t)
Instantiate a T in here from t.
- template<typename T >
T & [build](#) ()
- template<typename T >
T & [build](#) (const T &t)
- template<typename T >
T & [as](#) () [YY_NOEXCEPT](#)
Accessor to a built T.
- template<typename T >
const T & [as](#) () const [YY_NOEXCEPT](#)
Const accessor to a built T (for printer).
- template<typename T >
void [swap](#) ([self_type](#) &that) [YY_NOEXCEPT](#)
- template<typename T >
void [move](#) ([self_type](#) &that)
- template<typename T >
void [copy](#) (const [self_type](#) &that)
Copy the content of that to this.
- template<typename T >
void [destroy](#) ()
Destroy the stored T.

9.28.1 Detailed Description

A buffer to store and retrieve objects.

Sort of a variant, but does not keep track of the nature of the stored data, since that knowledge is available via the current parser state.

9.28.2 Member Typedef Documentation

9.28.2.1 self_type

```
typedef semantic_type annot::parser::semantic_type::self_type
```

Type of *this.

9.28.3 Constructor & Destructor Documentation

9.28.3.1 semantic_type() [1/2]

```
annot::parser::semantic_type::semantic_type ( ) [inline]
```

Empty construction.

9.28.3.2 semantic_type() [2/2]

```
template<typename T >  
annot::parser::semantic_type::semantic_type (   
    YY_RVREF(T) t ) [inline]
```

Construct and fill.

9.28.3.3 ~semantic_type()

```
annot::parser::semantic_type::~~semantic_type ( ) [inline]
```

Destruction, allowed only if empty.

9.28.4 Member Function Documentation

9.28.4.1 `as()` [1/2]

```
template<typename T >
const T& annot::parser::semantic_type::as ( ) const [inline]
```

Const accessor to a built *T* (for printer).

9.28.4.2 `as()` [2/2]

```
template<typename T >
T& annot::parser::semantic_type::as ( ) [inline]
```

Accessor to a built *T*.

9.28.4.3 `build()` [1/2]

```
template<typename T >
T& annot::parser::semantic_type::build ( ) [inline]
```

Instantiate an empty *T* in here. Obsolete, use `emplace`.

9.28.4.4 `build()` [2/2]

```
template<typename T >
T& annot::parser::semantic_type::build (
    const T & t ) [inline]
```

Instantiate a *T* in here from *t*. Obsolete, use `emplace`.

9.28.4.5 `copy()`

```
template<typename T >
void annot::parser::semantic_type::copy (
    const self_type & that ) [inline]
```

Copy the content of *that* to this.

9.28.4.6 destroy()

```
template<typename T >
void annot::parser::semantic_type::destroy ( ) [inline]
```

Destroy the stored *T*.

9.28.4.7 emplace() [1/2]

```
template<typename T >
T& annot::parser::semantic_type::emplace ( ) [inline]
```

Instantiate an empty *T* in here.

9.28.4.8 emplace() [2/2]

```
template<typename T >
T& annot::parser::semantic_type::emplace (
    const T & t ) [inline]
```

Instantiate a *T* in here from *t*.

9.28.4.9 move()

```
template<typename T >
void annot::parser::semantic_type::move (
    self_type & that ) [inline]
```

Move the content of *that* to this.

Destroys *that*.

9.28.4.10 swap()

```
template<typename T >
void annot::parser::semantic_type::swap (
    self_type & that ) [inline]
```

Swap the content with *that*, of same type.

Both variants must be built beforehand, because swapping the actual data requires reading it (with [as\(\)](#)), and this is not possible on unconstructed variants: it would require some dynamic testing, which should not be the variant's responsibility. Swapping between built and (possibly) non-built is done with [self_type::move\(\)](#).

9.28.5 Member Data Documentation

9.28.5.1 yyalalign_me

```
long double annot::parser::semantic_type::yyalign_me
```

Strongest alignment constraints.

9.28.5.2 yyraw

```
char annot::parser::semantic_type::yyraw[size]
```

A buffer large enough to store any of the semantic values.

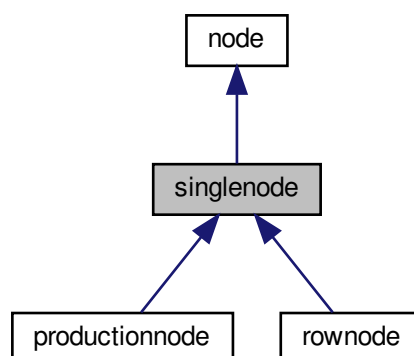
The documentation for this class was generated from the following file:

- [annot_parser.hh](#)

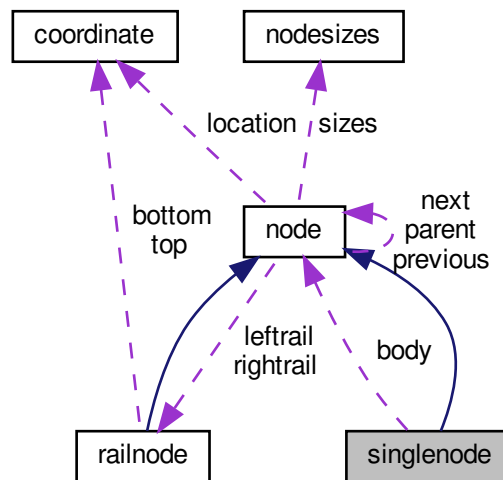
9.29 singlenode Class Reference

```
#include <graph.hh>
```

Inheritance diagram for singlenode:



Collaboration diagram for singlenode:



Public Member Functions

- [singlenode](#) ([node](#) *p)
- [singlenode](#) (const [singlenode](#) &original)
- virtual [singlenode](#) * [clone](#) () const
- virtual void [forgetChild](#) (int n)
- virtual void [drawToLeftRail](#) (ofstream &outs, [railnode](#) *p, [vraildir](#) join, int drawself)
- virtual void [drawToRightRail](#) (ofstream &outs, [railnode](#) *p, [vraildir](#) join, int drawself)
- virtual void [mergeRails](#) ()
- virtual ~[singlenode](#) ()
- virtual int [mergeConcats](#) (int depth)
- virtual int [liftConcats](#) (int depth)
- virtual int [mergeChoices](#) (int depth)
- virtual int [analyzeOptLoops](#) (int depth)
- virtual int [analyzeNonOptLoops](#) (int depth)
- virtual int [numChildren](#) ()
- virtual [node](#) * [getChild](#) (int n)
- virtual int [operator==](#) ([node](#) &r)
- virtual int [operator!=](#) ([node](#) &r)
- virtual [node](#) * [subsume](#) (regex_t *name, [node](#) *replacement)
- virtual void [setParent](#) ([node](#) *p)
- virtual void [setPrevious](#) ([node](#) *n)
- virtual void [setNext](#) ([node](#) *n)
- virtual void [fixSkips](#) ()
- virtual string [texName](#) ()

Protected Attributes

- [node](#) * [body](#)

Additional Inherited Members

9.29.1 Constructor & Destructor Documentation

9.29.1.1 `singlenode()` [1/2]

```
singlenode::singlenode (  
    node * p )
```

9.29.1.2 `singlenode()` [2/2]

```
singlenode::singlenode (  
    const singlenode & original )
```

9.29.1.3 `~singlenode()`

```
virtual singlenode::~~singlenode ( ) [inline], [virtual]
```

9.29.2 Member Function Documentation

9.29.2.1 `analyzeNonOptLoops()`

```
virtual int singlenode::analyzeNonOptLoops (  
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.2 `analyzeOptLoops()`

```
virtual int singlenode::analyzeOptLoops (  
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.3 clone()

```
singlenode * singlenode::clone ( ) const [virtual]
```

Implements [node](#).

Reimplemented in [productionnode](#), and [rownode](#).

9.29.2.4 drawToLeftRail()

```
void singlenode::drawToLeftRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.29.2.5 drawToRightRail()

```
void singlenode::drawToRightRail (
    ostream & outs,
    railnode * p,
    vraildir join,
    int drawself ) [virtual]
```

Reimplemented from [node](#).

9.29.2.6 fixSkips()

```
void singlenode::fixSkips ( ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [productionnode](#).

9.29.2.7 forgetChild()

```
void singlenode::forgetChild (
    int n ) [virtual]
```

Reimplemented from [node](#).

9.29.2.8 getChild()

```
virtual node* singlenode::getChild (
    int n ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.9 liftConcats()

```
int singlenode::liftConcats (
    int depth ) [virtual]
```

Reimplemented from [node](#).

9.29.2.10 mergeChoices()

```
virtual int singlenode::mergeChoices (
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.11 mergeConcats()

```
virtual int singlenode::mergeConcats (
    int depth ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.12 mergeRails()

```
virtual void singlenode::mergeRails ( ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.13 numChildren()

```
virtual int singlenode::numChildren ( ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.14 operator"!="()

```
virtual int singlenode::operator!= (
    node & r ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.15 operator==()

```
int singlenode::operator== (
    node & r ) [virtual]
```

Reimplemented from [node](#).

9.29.2.16 setNext()

```
virtual void singlenode::setNext (
    node * n ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.17 setParent()

```
void singlenode::setParent (
    node * p ) [virtual]
```

Reimplemented from [node](#).

9.29.2.18 setPrevious()

```
virtual void singlenode::setPrevious (
    node * n ) [inline], [virtual]
```

Reimplemented from [node](#).

9.29.2.19 subsume()

```
node * singlenode::subsume (
    regex_t * name,
    node * replacement ) [virtual]
```

Reimplemented from [node](#).

Reimplemented in [productionnode](#).

9.29.2.20 texName()

```
virtual string singlenode::texName ( ) [inline], [virtual]
```

Reimplemented from [node](#).

Reimplemented in [productionnode](#), and [rownode](#).

9.29.3 Member Data Documentation**9.29.3.1 body**

```
node* singlenode::body [protected]
```

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

- [graph.hh](#)
- [graph.cc](#)
- [optimize.cc](#)
- [output.cc](#)
- [subsume.cc](#)

9.30 annot::parser::stack< T, S >::slice Class Reference

Present a slice of the top of a stack.

```
#include <annot_parser.hh>
```

Public Member Functions

- [slice](#) (const stack &stack, index_type range)
- const T & [operator\[\]](#) (index_type i) const

9.30.1 Detailed Description

```
template<typename T, typename S = std::vector<T>>
class annot::parser::stack< T, S >::slice
```

Present a slice of the top of a stack.

9.30.2 Constructor & Destructor Documentation

9.30.2.1 slice()

```
template<typename T , typename S = std::vector<T>>
annot::parser::stack< T, S >::slice::slice (
    const stack & stack,
    index_type range ) [inline]
```

9.30.3 Member Function Documentation

9.30.3.1 operator[]()

```
template<typename T , typename S = std::vector<T>>
const T& annot::parser::stack< T, S >::slice::operator[] (
    index_type i ) const [inline]
```

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

- [annot_parser.hh](#)

9.31 yy::parser::stack< T, S >::slice Class Reference

Present a slice of the top of a stack.

```
#include <parser.hh>
```

Public Member Functions

- [slice](#) (const stack &stack, index_type range) [YY_NOEXCEPT](#)
- const T & [operator\[\]](#) (index_type i) const

9.31.1 Detailed Description

```
template<typename T, typename S = std::vector<T>>
class yy::parser::stack< T, S >::slice
```

Present a slice of the top of a stack.

9.31.2 Constructor & Destructor Documentation

9.31.2.1 slice()

```
template<typename T , typename S = std::vector<T>>
yy::parser::stack< T, S >::slice::slice (
    const stack & stack,
    index_type range ) [inline]
```

9.31.3 Member Function Documentation

9.31.3.1 operator[]()

```
template<typename T , typename S = std::vector<T>>
const T& yy::parser::stack< T, S >::slice::operator[] (
    index_type i ) const [inline]
```

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

- [parser.hh](#)

9.32 annot::parser::symbol_kind Struct Reference

Symbol kinds.

```
#include <annot_parser.hh>
```

Public Types

- enum [symbol_kind_type](#) {
[YYNTOKENS](#) = 12 , [S_YYEMPTY](#) = -2 , [S_YYEOF](#) = 0 , [S_YYerror](#) = 1 ,
[S_YYUNDEF](#) = 2 , [S_ASTART](#) = 3 , [S_AEND](#) = 4 , [S_SEMICOLON](#) = 5 ,
[S_SUBSUME](#) = 6 , [S_AS](#) = 7 , [S_CAPTION](#) = 8 , [S_SIDEWAYS](#) = 9 ,
[S_UNEXP](#) = 10 , [S_STRING](#) = 11 , [S_YYACCEPT](#) = 12 , [S_annotations](#) = 13 ,
[S_annots](#) = 14 , [S_annot](#) = 15 }

9.32.1 Detailed Description

Symbol kinds.

9.32.2 Member Enumeration Documentation

9.32.2.1 symbol_kind_type

```
enum annot::parser::symbol_kind::symbol_kind_type
```

Enumerator

YYNTOKENS	Number of tokens.
S_YYEMPTY	
S_YYEOF	
S_YYerror	
S_YYUNDEF	
S_ASTART	
S_AEND	
S_SEMICOLON	
S_SUBSUME	
S_AS	
S_CAPTION	
S_SIDEWAYS	
S_UNEXP	
S_STRING	
S_YYACCEPT	
S_annotations	
S_annots	
S_annot	

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

- [annot_parser.hh](#)

9.33 yy::parser::symbol_kind Struct Reference

Symbol kinds.

```
#include <parser.hh>
```

Public Types

- enum [symbol_kind_type](#) {
[YYNTOKENS](#) = 18 , [S_YYEMPTY](#) = -2 , [S_YYEOF](#) = 0 , [S_YYerror](#) = 1 ,
[S_YYUNDEF](#) = 2 , [S_COMMA](#) = 3 , [S_EQUAL](#) = 4 , [S_SEMICOLON](#) = 5 ,
[S_PIPE](#) = 6 , [S_LBRACK](#) = 7 , [S_RBRACK](#) = 8 , [S_LPAREN](#) = 9 ,
[S_RPAREN](#) = 10 , [S_LBRACE](#) = 11 , [S_RBRACE](#) = 12 , [S_NEWLINE](#) = 13 ,
[S_UNEXP](#) = 14 , [S_TERM](#) = 15 , [S_STRING](#) = 16 , [S_ANNOTATION](#) = 17 ,
[S_YYACCEPT](#) = 18 , [S_grammar](#) = 19 , [S_productions](#) = 20 , [S_production](#) = 21 ,
[S_annotations](#) = 22 , [S_rows](#) = 23 , [S_expression](#) = 24 , [S_primary](#) = 25 }

9.33.1 Detailed Description

Symbol kinds.

9.33.2 Member Enumeration Documentation

9.33.2.1 [symbol_kind_type](#)

enum [yy::parser::symbol_kind::symbol_kind_type](#)

Enumerator

YYNTOKENS	Number of tokens.
S_YYEMPTY	
S_YYEOF	
S_YYerror	
S_YYUNDEF	
S_COMMA	
S_EQUAL	
S_SEMICOLON	
S_PIPE	
S_LBRACK	
S_RBRACK	
S_LPAREN	
S_RPAREN	
S_LBRACE	
S_RBRACE	
S_NEWLINE	
S_UNEXP	
S_TERM	
S_STRING	
S_ANNOTATION	
S_YYACCEPT	
S_grammar	
S_productions	
S_production	
S_annotations	
S_rows	
S_expression	
S_primary	

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

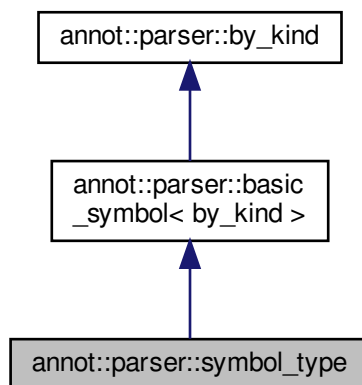
- [parser.hh](#)

9.34 annot::parser::symbol_type Struct Reference

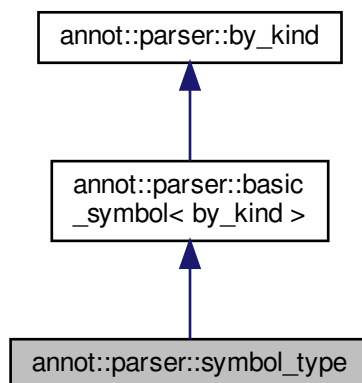
"External" symbols: returned by the scanner.

```
#include <annot_parser.hh>
```

Inheritance diagram for annot::parser::symbol_type:



Collaboration diagram for annot::parser::symbol_type:



Public Types

- typedef `basic_symbol< by_kind > super_type`
Superclass.

Public Member Functions

- `symbol_type ()`
Empty symbol.
- `symbol_type (int tok, const location_type &l)`
Constructor for valueless symbols, and symbols from each type.
- `symbol_type (int tok, const std::string &v, const location_type &l)`

Additional Inherited Members

9.34.1 Detailed Description

"External" symbols: returned by the scanner.

9.34.2 Member Typedef Documentation

9.34.2.1 `super_type`

```
typedef basic_symbol<by_kind> annot::parser::symbol_type::super_type
```

Superclass.

9.34.3 Constructor & Destructor Documentation

9.34.3.1 `symbol_type()` [1/3]

```
annot::parser::symbol_type::symbol_type ( ) [inline]
```

Empty symbol.

9.34.3.2 symbol_type() [2/3]

```
annot::parser::symbol_type::symbol_type (
    int tok,
    const location_type & l ) [inline]
```

Constructor for valueless symbols, and symbols from each type.

9.34.3.3 symbol_type() [3/3]

```
annot::parser::symbol_type::symbol_type (
    int tok,
    const std::string & v,
    const location_type & l ) [inline]
```

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

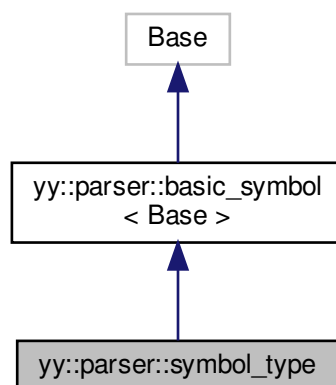
- [annot_parser.hh](#)

9.35 yy::parser::symbol_type Struct Reference

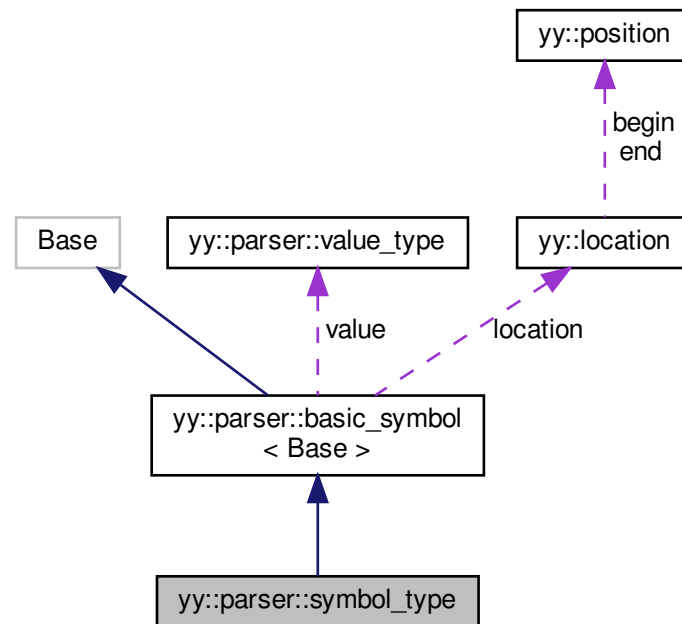
"External" symbols: returned by the scanner.

```
#include <parser.hh>
```

Inheritance diagram for yy::parser::symbol_type:



Collaboration diagram for `yy::parser::symbol_type`:



Public Types

- typedef `basic_symbol< by_kind > super_type`
Superclass.

Public Member Functions

- `symbol_type () YY_NOEXCEPT`
Empty symbol.
- `symbol_type (int tok, const location_type &l)`
Constructor for valueless symbols, and symbols from each type.
- `symbol_type (int tok, const std::string &v, const location_type &l)`

Additional Inherited Members

9.35.1 Detailed Description

"External" symbols: returned by the scanner.

9.35.2 Member Typedef Documentation

9.35.2.1 super_type

```
typedef basic_symbol<by_kind> yy::parser::symbol_type::super_type
```

Superclass.

9.35.3 Constructor & Destructor Documentation

9.35.3.1 symbol_type() [1/3]

```
yy::parser::symbol_type::symbol_type ( ) [inline]
```

Empty symbol.

9.35.3.2 symbol_type() [2/3]

```
yy::parser::symbol_type::symbol_type (
    int tok,
    const location_type & l ) [inline]
```

Constructor for valueless symbols, and symbols from each type.

9.35.3.3 symbol_type() [3/3]

```
yy::parser::symbol_type::symbol_type (
    int tok,
    const std::string & v,
    const location_type & l ) [inline]
```

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

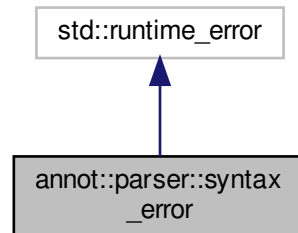
- [parser.hh](#)

9.36 annot::parser::syntax_error Struct Reference

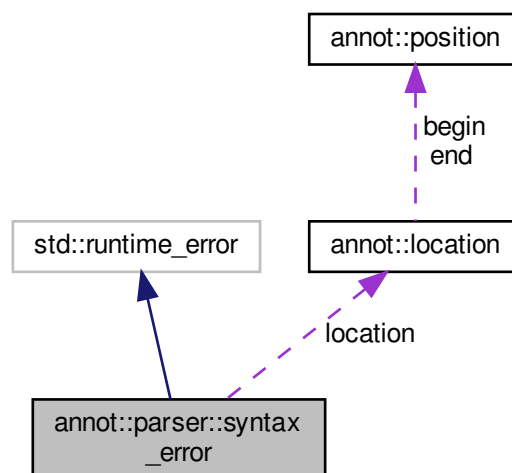
Syntax errors thrown from user actions.

```
#include <annot_parser.hh>
```

Inheritance diagram for `annot::parser::syntax_error`:



Collaboration diagram for `annot::parser::syntax_error`:



Public Member Functions

- [syntax_error](#) (const [location_type](#) &l, const std::string &m)
- [syntax_error](#) (const [syntax_error](#) &s)
- [~syntax_error](#) () YY_NOEXCEPT YY_NOTHROW

Public Attributes

- [location_type](#) [location](#)

9.36.1 Detailed Description

Syntax errors thrown from user actions.

9.36.2 Constructor & Destructor Documentation

9.36.2.1 [syntax_error\(\)](#) [1/2]

```
annot::parser::syntax_error::syntax_error (
    const location\_type & l,
    const std::string & m ) [inline]
```

9.36.2.2 [syntax_error\(\)](#) [2/2]

```
annot::parser::syntax_error::syntax_error (
    const syntax\_error & s ) [inline]
```

9.36.2.3 [~syntax_error\(\)](#)

```
annot::parser::syntax_error::~~syntax_error ( )
```

9.36.3 Member Data Documentation

9.36.3.1 [location](#)

[location_type](#) [annot::parser::syntax_error::location](#)

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

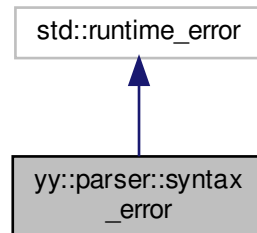
- [annot_parser.hh](#)
- [annot_parser.cc](#)

9.37 yy::parser::syntax_error Struct Reference

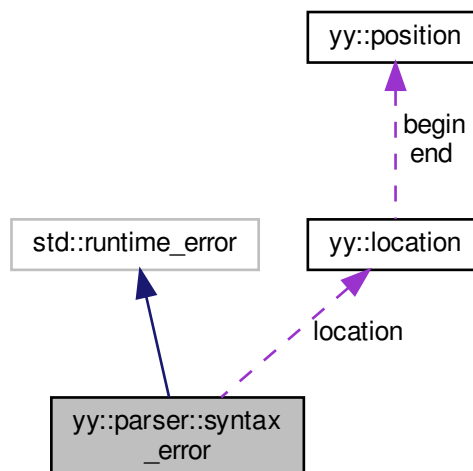
Syntax errors thrown from user actions.

```
#include <parser.hh>
```

Inheritance diagram for yy::parser::syntax_error:



Collaboration diagram for yy::parser::syntax_error:



Public Member Functions

- [syntax_error](#) (const [location_type](#) &l, const std::string &m)
- [syntax_error](#) (const [syntax_error](#) &s)
- [~syntax_error](#) () YY_NOEXCEPT YY_NOTHROW

Public Attributes

- [location_type](#) [location](#)

9.37.1 Detailed Description

Syntax errors thrown from user actions.

9.37.2 Constructor & Destructor Documentation

9.37.2.1 `syntax_error()` [1/2]

```
yy::parser::syntax_error::syntax_error (
    const location\_type & l,
    const std::string & m ) [inline]
```

9.37.2.2 `syntax_error()` [2/2]

```
yy::parser::syntax_error::syntax_error (
    const syntax\_error & s ) [inline]
```

9.37.2.3 `~syntax_error()`

```
yy::parser::syntax_error::~~syntax_error ( )
```

9.37.3 Member Data Documentation

9.37.3.1 `location`

[location_type](#) yy::parser::syntax_error::location

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

- [parser.hh](#)
- [parser.cc](#)

Public Member Functions

- [termnode](#) (string s)
- [termnode](#) (const [termnode](#) &original)
- virtual [termnode](#) * [clone](#) () const
- virtual [~termnode](#) ()

Additional Inherited Members

9.38.1 Constructor & Destructor Documentation

9.38.1.1 [termnode\(\)](#) [1/2]

```
termnode::termnode (  
    string s )
```

9.38.1.2 [termnode\(\)](#) [2/2]

```
termnode::termnode (  
    const termnode & original )
```

9.38.1.3 [~termnode\(\)](#)

```
virtual termnode::~~termnode ( ) [inline], [virtual]
```

9.38.2 Member Function Documentation

9.38.2.1 [clone\(\)](#)

```
termnode * termnode::clone ( ) const [virtual]
```

Reimplemented from [nontermnode](#).

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

- [graph.hh](#)
- [graph.cc](#)

9.39 annot::parser::token Struct Reference

Token kinds.

```
#include <annot_parser.hh>
```

Public Types

- enum [token_kind_type](#) {
[TOK_ANNOTEMPTY](#) = -2 , [TOK_END](#) = 0 , [TOK_ANNOTerror](#) = 1 , [TOK_ANNOTUNDEF](#) = 2 ,
[TOK_ASTART](#) = 3 , [TOK_AEND](#) = 4 , [TOK_SEMICOLON](#) = 5 , [TOK_SUBSUME](#) = 6 ,
[TOK_AS](#) = 7 , [TOK_CAPTION](#) = 8 , [TOK_SIDEWAYS](#) = 9 , [TOK_UNEXP](#) = 10 ,
[TOK_STRING](#) = 11 }
- typedef [token_kind_type](#) [yytokentype](#)
Backward compatibility alias (Bison 3.6).

9.39.1 Detailed Description

Token kinds.

9.39.2 Member Typedef Documentation

9.39.2.1 yytokentype

```
typedef token\_kind\_type annot::parser::token::yytokentype
```

Backward compatibility alias (Bison 3.6).

9.39.3 Member Enumeration Documentation

9.39.3.1 token_kind_type

```
enum annot::parser::token::token\_kind\_type
```

Enumerator

TOK_ANNOTEMPTY	
TOK_END	
TOK_ANNOTerror	
TOK_ANNOTUNDEF	
TOK_ASTART	
TOK_AEND	
TOK_SEMICOLON	
TOK_SUBSUME	
TOK_AS	
TOK_CAPTION	

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

- [annot_parser.hh](#)

9.40 yy::parser::token Struct Reference

Token kinds.

```
#include <parser.hh>
```

Public Types

- enum [token_kind_type](#) {
 [TOK_YEMPTY](#) = -2 , [TOK_END](#) = 0 , [TOK_YYerror](#) = 1 , [TOK_YYUNDEF](#) = 2 ,
 [TOK_COMMA](#) = 3 , [TOK_EQUAL](#) = 4 , [TOK_SEMICOLON](#) = 5 , [TOK_PIPE](#) = 6 ,
 [TOK_LBRACK](#) = 7 , [TOK_RBRACK](#) = 8 , [TOK_LPAREN](#) = 9 , [TOK_RPAREN](#) = 10 ,
 [TOK_LBRACE](#) = 11 , [TOK_RBRACE](#) = 12 , [TOK_NEWLINE](#) = 13 , [TOK_UNEXP](#) = 14 ,
 [TOK_TERM](#) = 15 , [TOK_STRING](#) = 16 , [TOK_ANNOTATION](#) = 17 }
• typedef [token_kind_type](#) [yytokentype](#)
 Backward compatibility alias (Bison 3.6).

9.40.1 Detailed Description

Token kinds.

9.40.2 Member Typedef Documentation

9.40.2.1 yytokentype

```
typedef token\_kind\_type yy::parser::token::yytokentype
```

Backward compatibility alias (Bison 3.6).

9.40.3 Member Enumeration Documentation

9.40.3.1 token_kind_type

```
enum yy::parser::token::token\_kind\_type
```

Enumerator

TOK_YYEMPTY	
TOK_END	
TOK_YYerror	
TOK_YYUNDEF	
TOK_COMMA	
TOK_EQUAL	
TOK_SEMICOLON	
TOK_PIPE	
TOK_LBRACK	
TOK_RBRACK	
TOK_LPAREN	
TOK_RPAREN	
TOK_LBRACE	
TOK_RBRACE	
TOK_NEWLINE	
TOK_UNEXP	
TOK_TERM	
TOK_STRING	
TOK_ANNOTATION	

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

- [parser.hh](#)

9.41 yy::parser::value_type Class Reference

```
#include <parser.hh>
```

Public Types

- typedef [value_type](#) [self_type](#)
*Type of *this.*

Public Member Functions

- [value_type](#) () [YY_NOEXCEPT](#)
Empty construction.
- template<typename T >
[value_type](#) ([YY_RVREF](#)(T) t)
Construct and fill.
- [~value_type](#) () [YY_NOEXCEPT](#)
Destruction, allowed only if empty.
- template<typename T >
T & [emplace](#) ()
Instantiate an empty T in here.

- `template<typename T >`
`T & emplace (const T &t)`
Instantiate a T in here from t.
- `template<typename T >`
`T & build ()`
- `template<typename T >`
`T & build (const T &t)`
- `template<typename T >`
`T & as () YY_NOEXCEPT`
Accessor to a built T.
- `template<typename T >`
`const T & as () const YY_NOEXCEPT`
Const accessor to a built T (for printer).
- `template<typename T >`
`void swap (self_type &that) YY_NOEXCEPT`
- `template<typename T >`
`void move (self_type &that)`
- `template<typename T >`
`void copy (const self_type &that)`
Copy the content of that to this.
- `template<typename T >`
`void destroy ()`
Destroy the stored T.

9.41.1 Detailed Description

A buffer to store and retrieve objects.

Sort of a variant, but does not keep track of the nature of the stored data, since that knowledge is available via the current parser state.

9.41.2 Member Typedef Documentation

9.41.2.1 `self_type`

```
typedef value\_type yy::parser::value_type::self_type
```

Type of `*this`.

9.41.3 Constructor & Destructor Documentation

9.41.3.1 value_type() [1/2]

```
yy::parser::value_type::value_type ( ) [inline]
```

Empty construction.

9.41.3.2 value_type() [2/2]

```
template<typename T >  
yy::parser::value_type::value_type (   
    YY_RVREF(T) t ) [inline]
```

Construct and fill.

9.41.3.3 ~value_type()

```
yy::parser::value_type::~~value_type ( ) [inline]
```

Destruction, allowed only if empty.

9.41.4 Member Function Documentation

9.41.4.1 as() [1/2]

```
template<typename T >  
const T& yy::parser::value_type::as ( ) const [inline]
```

Const accessor to a built *T* (for printer).

9.41.4.2 as() [2/2]

```
template<typename T >  
T& yy::parser::value_type::as ( ) [inline]
```

Accessor to a built *T*.

9.41.4.3 build() [1/2]

```
template<typename T >
T& yy::parser::value_type::build ( ) [inline]
```

Instantiate an empty *T* in here. Obsolete, use `emplace`.

9.41.4.4 build() [2/2]

```
template<typename T >
T& yy::parser::value_type::build (
    const T & t ) [inline]
```

Instantiate a *T* in here from *t*. Obsolete, use `emplace`.

9.41.4.5 copy()

```
template<typename T >
void yy::parser::value_type::copy (
    const self_type & that ) [inline]
```

Copy the content of *that* to this.

9.41.4.6 destroy()

```
template<typename T >
void yy::parser::value_type::destroy ( ) [inline]
```

Destroy the stored *T*.

9.41.4.7 emplace() [1/2]

```
template<typename T >
T& yy::parser::value_type::emplace ( ) [inline]
```

Instantiate an empty *T* in here.

9.41.4.8 emplace() [2/2]

```
template<typename T >
T& yy::parser::value_type::emplace (
    const T & t ) [inline]
```

Instantiate a *T* in here from *t*.

9.41.4.9 move()

```
template<typename T >
void yy::parser::value_type::move (
    self_type & that ) [inline]
```

Move the content of *that* to this.

Destroys *that*.

9.41.4.10 swap()

```
template<typename T >
void yy::parser::value_type::swap (
    self_type & that ) [inline]
```

Swap the content with *that*, of same type.

Both variants must be built beforehand, because swapping the actual data requires reading it (with [as\(\)](#)), and this is not possible on unconstructed variants: it would require some dynamic testing, which should not be the variant's responsibility. Swapping between built and (possibly) non-built is done with [self_type::move](#) ().

9.41.5 Member Data Documentation

9.41.5.1 yyalign_me_

```
long double yy::parser::value_type::yyalign_me_
```

Strongest alignment constraints.

9.41.5.2 yyraw_

```
char yy::parser::value_type::yyraw_[size]
```

A buffer large enough to store any of the semantic values.

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

- [parser.hh](#)

9.42 yy_buffer_state Struct Reference

```
#include <annot_lexer.hh>
```

Public Attributes

- FILE * [yy_input_file](#)
- char * [yy_ch_buf](#)
- char * [yy_buf_pos](#)
- int [yy_buf_size](#)
- int [yy_n_chars](#)
- int [yy_is_our_buffer](#)
- int [yy_is_interactive](#)
- int [yy_at_bol](#)
- int [yy_bs_lineno](#)
- int [yy_bs_column](#)
- int [yy_fill_buffer](#)
- int [yy_buffer_status](#)

9.42.1 Member Data Documentation

9.42.1.1 yy_at_bol

```
int yy_buffer_state::yy_at_bol
```

9.42.1.2 yy_bs_column

```
int yy_buffer_state::yy_bs_column
```

The column count.

9.42.1.3 yy_bs_lineno

```
int yy_buffer_state::yy_bs_lineno
```

The line count.

9.42.1.4 yy_buf_pos

```
char * yy_buffer_state::yy_buf_pos
```

9.42.1.5 yy_buf_size

```
int yy_buffer_state::yy_buf_size
```

9.42.1.6 yy_buffer_status

```
int yy_buffer_state::yy_buffer_status
```

9.42.1.7 yy_ch_buf

```
char * yy_buffer_state::yy_ch_buf
```

9.42.1.8 yy_fill_buffer

```
int yy_buffer_state::yy_fill_buffer
```

9.42.1.9 yy_input_file

```
FILE * yy_buffer_state::yy_input_file
```

9.42.1.10 yy_is_interactive

```
int yy_buffer_state::yy_is_interactive
```

9.42.1.11 yy_is_our_buffer

```
int yy_buffer_state::yy_is_our_buffer
```

9.42.1.12 yy_n_chars

```
int yy_buffer_state::yy_n_chars
```

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

- [annot_lexer.cc](#)
- [annot_lexer.hh](#)
- [lexer.cc](#)
- [lexer.hh](#)

9.43 yy_trans_info Struct Reference

Public Attributes

- [flex_int32_t yy_verify](#)
- [flex_int32_t yy_nxt](#)

9.43.1 Member Data Documentation

9.43.1.1 yy_nxt

[flex_int32_t](#) yy_trans_info::yy_nxt

9.43.1.2 yy_verify

[flex_int32_t](#) yy_trans_info::yy_verify

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

- [annot_lexer.cc](#)
- [lexer.cc](#)

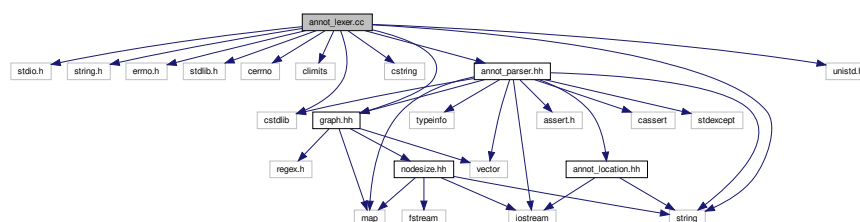
Chapter 10

File Documentation

10.1 annot_lexer.cc File Reference

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <cerrno>
#include <climits>
#include <cstdlib>
#include <cstring>
#include <string>
#include "graph.hh"
#include "annot_parser.hh"
#include <unistd.h>
```

Include dependency graph for annot_lexer.cc:



Classes

- struct [yy_buffer_state](#)
- struct [yy_trans_info](#)

Macros

- #define [YY_INT_ALIGNED](#) short int
- #define [yy_create_buffer](#) annot_create_buffer
- #define [yy_delete_buffer](#) annot_delete_buffer
- #define [yy_scan_buffer](#) annot_scan_buffer
- #define [yy_scan_string](#) annot_scan_string
- #define [yy_scan_bytes](#) annot_scan_bytes
- #define [yy_init_buffer](#) annot_init_buffer
- #define [yy_flush_buffer](#) annot_flush_buffer
- #define [yy_load_buffer_state](#) annot_load_buffer_state
- #define [yy_switch_to_buffer](#) annot_switch_to_buffer
- #define [yypush_buffer_state](#) annotpush_buffer_state
- #define [yypop_buffer_state](#) annotpop_buffer_state
- #define [yyensure_buffer_stack](#) annotensure_buffer_stack
- #define [yy_flex_debug](#) annot_flex_debug
- #define [yyin](#) annotin
- #define [yyleng](#) annotleng
- #define [yylex](#) annotlex
- #define [yylineno](#) annotlineno
- #define [yyout](#) annotout
- #define [yyrestart](#) annotrestart
- #define [yytext](#) annottext
- #define [yywrap](#) annotwrap
- #define [yyalloc](#) annotalloc
- #define [yyrealloc](#) annotrealloc
- #define [yyfree](#) annotfree
- #define [FLEX_SCANNER](#)
- #define [YY_FLEX_MAJOR_VERSION](#) 2
- #define [YY_FLEX_MINOR_VERSION](#) 6
- #define [YY_FLEX_SUBMINOR_VERSION](#) 4
- #define [FLEX_BETA](#)
- #define [annot_create_buffer_ALREADY_DEFINED](#)
- #define [annot_delete_buffer_ALREADY_DEFINED](#)
- #define [annot_scan_buffer_ALREADY_DEFINED](#)
- #define [annot_scan_string_ALREADY_DEFINED](#)
- #define [annot_scan_bytes_ALREADY_DEFINED](#)
- #define [annot_init_buffer_ALREADY_DEFINED](#)
- #define [annot_flush_buffer_ALREADY_DEFINED](#)
- #define [annot_load_buffer_state_ALREADY_DEFINED](#)
- #define [annot_switch_to_buffer_ALREADY_DEFINED](#)
- #define [annotpush_buffer_state_ALREADY_DEFINED](#)
- #define [annotpop_buffer_state_ALREADY_DEFINED](#)
- #define [annotensure_buffer_stack_ALREADY_DEFINED](#)
- #define [annotlex_ALREADY_DEFINED](#)
- #define [annotrestart_ALREADY_DEFINED](#)
- #define [yylex_init](#) annotlex_init
- #define [yylex_init_extra](#) annotlex_init_extra
- #define [yylex_destroy](#) annotlex_destroy
- #define [yyget_debug](#) annotget_debug
- #define [yyset_debug](#) annotset_debug
- #define [yyget_extra](#) annotget_extra
- #define [yyset_extra](#) annotset_extra
- #define [yyget_in](#) annotget_in
- #define [yyset_in](#) annotset_in

- #define [yyget_out](#) annotget_out
- #define [yyset_out](#) annotset_out
- #define [yyget_leng](#) annotget_leng
- #define [yyget_text](#) annotget_text
- #define [yyget_lineno](#) annotget_lineno
- #define [yyset_lineno](#) annotset_lineno
- #define [annotwrap_ALREADY_DEFINED](#)
- #define [annotalloc_ALREADY_DEFINED](#)
- #define [annotrealloc_ALREADY_DEFINED](#)
- #define [annotfree_ALREADY_DEFINED](#)
- #define [annottext_ALREADY_DEFINED](#)
- #define [annotleng_ALREADY_DEFINED](#)
- #define [annotin_ALREADY_DEFINED](#)
- #define [annotout_ALREADY_DEFINED](#)
- #define [annot_flex_debug_ALREADY_DEFINED](#)
- #define [annotlineno_ALREADY_DEFINED](#)
- #define [FLEXINT_H](#)
- #define [INT8_MIN](#) (-128)
- #define [INT16_MIN](#) (-32767-1)
- #define [INT32_MIN](#) (-2147483647-1)
- #define [INT8_MAX](#) (127)
- #define [INT16_MAX](#) (32767)
- #define [INT32_MAX](#) (2147483647)
- #define [UINT8_MAX](#) (255U)
- #define [UINT16_MAX](#) (65535U)
- #define [UINT32_MAX](#) (4294967295U)
- #define [SIZE_MAX](#) (~(size_t)0)
- #define [yyconst](#) const
- #define [ynoreturn](#)
- #define [YY_NULL](#) 0
- #define [YY_SC_TO_UI](#)(c) (([YY_CHAR](#)) (c))
- #define [BEGIN](#) (yy_start) = 1 + 2 *
- #define [YY_START](#) (((yy_start) - 1) / 2)
- #define [YYSTATE](#) [YY_START](#)
- #define [YY_STATE_EOF](#)(state) ([YY_END_OF_BUFFER](#) + state + 1)
- #define [YY_NEW_FILE](#) yyrestart([yyin](#))
- #define [YY_END_OF_BUFFER_CHAR](#) 0
- #define [YY_BUF_SIZE](#) 16384
- #define [YY_STATE_BUF_SIZE](#) (([YY_BUF_SIZE](#) + 2) * sizeof([yy_state_type](#)))
- #define [YY_TYPEDEF_Y_BUFFER_STATE](#)
- #define [YY_TYPEDEF_Y_SIZE_T](#)
- #define [EOB_ACT_CONTINUE_SCAN](#) 0
- #define [EOB_ACT_END_OF_FILE](#) 1
- #define [EOB_ACT_LAST_MATCH](#) 2
- #define [YY_LESS_LINENO](#)(n)
- #define [YY_LINENO_REWIND_TO](#)(ptr)
- #define [yyless](#)(n)
- #define [unput](#)(c) yyunput(c, ([yytext_ptr](#)))
- #define [YY_STRUCT_Y_BUFFER_STATE](#)
- #define [YY_BUFFER_NEW](#) 0
- #define [YY_BUFFER_NORMAL](#) 1
- #define [YY_BUFFER_EOF_PENDING](#) 2
- #define [YY_CURRENT_BUFFER](#)
- #define [YY_CURRENT_BUFFER_LVALUE](#) (yy_buffer_stack)[(yy_buffer_stack_top)]
- #define [YY_FLUSH_BUFFER](#) yy_flush_buffer([YY_CURRENT_BUFFER](#))

- `#define yy_new_buffer yy_create_buffer`
- `#define yy_set_interactive(is_interactive)`
- `#define yy_set_bol(at_bol)`
- `#define YY_AT_BOL() (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)`
- `#define annotwrap() (/*CONSTCOND*/1)`
- `#define YY_SKIP_YYWRAP`
- `#define yytext_ptr yytext`
- `#define YY_DO_BEFORE_ACTION`
- `#define YY_NUM_RULES 15`
- `#define YY_END_OF_BUFFER 16`
- `#define REJECT reject_used_but_not_detected`
- `#define yymore() yymore_used_but_not_detected`
- `#define YY_MORE_ADJ 0`
- `#define YY_RESTORE_YY_MORE_OFFSET`
- `#define YY_DECL annot::parser::symbol_type annotlex (annotmap *m)`
- `#define YY_NO_INPUT 1`
- `#define YY_USER_ACTION aloc.columns (yyleng);`
- `#define INITIAL 0`
- `#define YY_EXTRA_TYPE void *`
- `#define YY_READ_BUF_SIZE 8192`
- `#define ECHO do { if (fwrite(yytext, (size_t) yyleng, 1, yyout)) {} } while (0)`
- `#define YY_INPUT(buf, result, max_size)`
- `#define yyterminate() return YY_NULL`
- `#define YY_START_STACK_INCR 25`
- `#define YY_FATAL_ERROR(msg) yy_fatal_error(msg)`
- `#define YY_BREAK /*LINTED*/break;`
- `#define YY_RULE_SETUP YY_USER_ACTION`
- `#define YY_EXIT_FAILURE 2`
- `#define yyless(n)`
- `#define YYTABLES_NAME "yytables"`

Typedefs

- `typedef signed char flex_int8_t`
- `typedef short int flex_int16_t`
- `typedef int flex_int32_t`
- `typedef unsigned char flex_uint8_t`
- `typedef unsigned short int flex_uint16_t`
- `typedef unsigned int flex_uint32_t`
- `typedef struct yy_buffer_state * YY_BUFFER_STATE`
- `typedef size_t yy_size_t`
- `typedef flex_uint8_t YY_CHAR`
- `typedef int yy_state_type`

Functions

- void [yyrestart](#) (FILE *input_file)
- void [yy_switch_to_buffer](#) (YY_BUFFER_STATE new_buffer)
- YY_BUFFER_STATE [yy_create_buffer](#) (FILE *file, int size)
- void [yy_delete_buffer](#) (YY_BUFFER_STATE b)
- void [yy_flush_buffer](#) (YY_BUFFER_STATE b)
- void [yypush_buffer_state](#) (YY_BUFFER_STATE new_buffer)
- YY_BUFFER_STATE [yy_scan_buffer](#) (char *base, [yy_size_t](#) size)
- YY_BUFFER_STATE [yy_scan_string](#) (const char *yy_str)
- YY_BUFFER_STATE [yy_scan_bytes](#) (const char *bytes, int len)
- void * [yyalloc](#) ([yy_size_t](#))
- void * [yyrealloc](#) (void *, [yy_size_t](#))
- void [yyfree](#) (void *)
- string [stripquotes](#) (string s)
- void [yyset_debug](#) (int debug_flag)
- void [yyset_extra](#) (YY_EXTRA_TYPE user_defined)
- void [yyset_in](#) (FILE *_in_str)
- void [yyset_out](#) (FILE *_out_str)
- void [yyset_lineno](#) (int _line_number)
- if (!(yy_init))
- [aloc step](#) ()
- [while](#) (1)

Variables

- int [yyleng](#)
- FILE * [yyin](#) = NULL
- FILE * [yyout](#) = NULL
- int [yylineno](#) = 1
- char * [yytext](#)
- int [yy_flex_debug](#) = 0
- YY_DECL
- char * [yy_cp](#)
- char * [yy_bp](#)
- int [yy_act](#)

10.1.1 Macro Definition Documentation

10.1.1.1 annot_create_buffer_ALREADY_DEFINED

```
#define annot_create_buffer_ALREADY_DEFINED
```

10.1.1.2 annot_delete_buffer_ALREADY_DEFINED

```
#define annot_delete_buffer_ALREADY_DEFINED
```

10.1.1.3 `annot_flex_debug_ALREADY_DEFINED`

```
#define annot_flex_debug_ALREADY_DEFINED
```

10.1.1.4 `annot_flush_buffer_ALREADY_DEFINED`

```
#define annot_flush_buffer_ALREADY_DEFINED
```

10.1.1.5 `annot_init_buffer_ALREADY_DEFINED`

```
#define annot_init_buffer_ALREADY_DEFINED
```

10.1.1.6 `annot_load_buffer_state_ALREADY_DEFINED`

```
#define annot_load_buffer_state_ALREADY_DEFINED
```

10.1.1.7 `annot_scan_buffer_ALREADY_DEFINED`

```
#define annot_scan_buffer_ALREADY_DEFINED
```

10.1.1.8 `annot_scan_bytes_ALREADY_DEFINED`

```
#define annot_scan_bytes_ALREADY_DEFINED
```

10.1.1.9 `annot_scan_string_ALREADY_DEFINED`

```
#define annot_scan_string_ALREADY_DEFINED
```

10.1.1.10 `annot_switch_to_buffer_ALREADY_DEFINED`

```
#define annot_switch_to_buffer_ALREADY_DEFINED
```


10.1.1.11 annotalloc_ALREADY_DEFINED

```
#define annotalloc_ALREADY_DEFINED
```

10.1.1.12 annotensure_buffer_stack_ALREADY_DEFINED

```
#define annotensure_buffer_stack_ALREADY_DEFINED
```

10.1.1.13 annotfree_ALREADY_DEFINED

```
#define annotfree_ALREADY_DEFINED
```

10.1.1.14 annotin_ALREADY_DEFINED

```
#define annotin_ALREADY_DEFINED
```

10.1.1.15 annotleng_ALREADY_DEFINED

```
#define annotleng_ALREADY_DEFINED
```

10.1.1.16 annotlex_ALREADY_DEFINED

```
#define annotlex_ALREADY_DEFINED
```

10.1.1.17 annotlineno_ALREADY_DEFINED

```
#define annotlineno_ALREADY_DEFINED
```

10.1.1.18 annotout_ALREADY_DEFINED

```
#define annotout_ALREADY_DEFINED
```

10.1.1.19 annotpop_buffer_state_ALREADY_DEFINED

```
#define annotpop_buffer_state_ALREADY_DEFINED
```

10.1.1.20 annotpush_buffer_state_ALREADY_DEFINED

```
#define annotpush_buffer_state_ALREADY_DEFINED
```

10.1.1.21 annotrealloc_ALREADY_DEFINED

```
#define annotrealloc_ALREADY_DEFINED
```

10.1.1.22 annotrestart_ALREADY_DEFINED

```
#define annotrestart_ALREADY_DEFINED
```

10.1.1.23 annottext_ALREADY_DEFINED

```
#define annottext_ALREADY_DEFINED
```

10.1.1.24 annotwrap

```
#define annotwrap( ) (/*CONSTCOND*/1)
```

10.1.1.25 annotwrap_ALREADY_DEFINED

```
#define annotwrap_ALREADY_DEFINED
```

10.1.1.26 BEGIN

```
#define BEGIN (yy_start) = 1 + 2 *
```

10.1.1.27 ECHO

```
#define ECHO do { if (fwrite( yytext, (size_t) yylen, 1, yyout )) {} } while (0)
```

10.1.1.28 EOB_ACT_CONTINUE_SCAN

```
#define EOB_ACT_CONTINUE_SCAN 0
```

10.1.1.29 EOB_ACT_END_OF_FILE

```
#define EOB_ACT_END_OF_FILE 1
```

10.1.1.30 EOB_ACT_LAST_MATCH

```
#define EOB_ACT_LAST_MATCH 2
```

10.1.1.31 FLEX_BETA

```
#define FLEX_BETA
```

10.1.1.32 FLEX_SCANNER

```
#define FLEX_SCANNER
```

10.1.1.33 FLEXINT_H

```
#define FLEXINT_H
```

10.1.1.34 INITIAL

```
#define INITIAL 0
```

10.1.1.35 INT16_MAX

```
#define INT16_MAX (32767)
```

10.1.1.36 INT16_MIN

```
#define INT16_MIN (-32767-1)
```

10.1.1.37 INT32_MAX

```
#define INT32_MAX (2147483647)
```

10.1.1.38 INT32_MIN

```
#define INT32_MIN (-2147483647-1)
```

10.1.1.39 INT8_MAX

```
#define INT8_MAX (127)
```

10.1.1.40 INT8_MIN

```
#define INT8_MIN (-128)
```

10.1.1.41 REJECT

```
#define REJECT reject_used_but_not_detected
```

10.1.1.42 SIZE_MAX

```
#define SIZE_MAX (~(size_t)0)
```

10.1.1.43 UINT16_MAX

```
#define UINT16_MAX (65535U)
```

10.1.1.44 UINT32_MAX

```
#define UINT32_MAX (4294967295U)
```

10.1.1.45 UINT8_MAX

```
#define UINT8_MAX (255U)
```

10.1.1.46 unput

```
#define unput(  
    c ) yyunput( c, (yytext_ptr) )
```

10.1.1.47 YY_AT_BOL

```
#define YY_AT_BOL( ) (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)
```

10.1.1.48 YY_BREAK

```
#define YY_BREAK /*LINTED*/break;
```

10.1.1.49 YY_BUF_SIZE

```
#define YY_BUF_SIZE 16384
```

10.1.1.50 YY_BUFFER_EOF_PENDING

```
#define YY_BUFFER_EOF_PENDING 2
```

10.1.1.51 YY_BUFFER_NEW

```
#define YY_BUFFER_NEW 0
```

10.1.1.52 YY_BUFFER_NORMAL

```
#define YY_BUFFER_NORMAL 1
```

10.1.1.53 yy_create_buffer

```
#define yy_create_buffer annot_create_buffer
```

10.1.1.54 YY_CURRENT_BUFFER

```
#define YY_CURRENT_BUFFER
```

Value:

```
( (yy_buffer_stack) \
? (yy_buffer_stack)[(yy_buffer_stack_top)] \
: NULL)
```

10.1.1.55 YY_CURRENT_BUFFER_LVALUE

```
#define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]
```

10.1.1.56 YY_DECL

```
#define YY_DECL annot::parser::symbol_type annotlex (annotmap *m)
```

10.1.1.57 yy_delete_buffer

```
#define yy_delete_buffer annot_delete_buffer
```

10.1.1.58 YY_DO_BEFORE_ACTION

```
#define YY_DO_BEFORE_ACTION
```

Value:

```
(yytext_ptr) = yy_bp; \  
yyleng = (int) (yy_cp - yy_bp); \  
(yy_hold_char) = *yy_cp; \  
*yy_cp = '\0'; \  
(yy_c_buf_p) = yy_cp;
```

10.1.1.59 YY_END_OF_BUFFER

```
#define YY_END_OF_BUFFER 16
```

10.1.1.60 YY_END_OF_BUFFER_CHAR

```
#define YY_END_OF_BUFFER_CHAR 0
```

10.1.1.61 YY_EXIT_FAILURE

```
#define YY_EXIT_FAILURE 2
```

10.1.1.62 YY_EXTRA_TYPE

```
#define YY_EXTRA_TYPE void *
```

10.1.1.63 YY_FATAL_ERROR

```
#define YY_FATAL_ERROR(  
    msg ) yy_fatal_error( msg )
```

10.1.1.64 yy_flex_debug

```
int yy_flex_debug annot_flex_debug
```

10.1.1.65 YY_FLEX_MAJOR_VERSION

```
#define YY_FLEX_MAJOR_VERSION 2
```

10.1.1.66 YY_FLEX_MINOR_VERSION

```
#define YY_FLEX_MINOR_VERSION 6
```

10.1.1.67 YY_FLEX_SUBMINOR_VERSION

```
#define YY_FLEX_SUBMINOR_VERSION 4
```

10.1.1.68 yy_flush_buffer

```
#define yy_flush_buffer annot_flush_buffer
```

10.1.1.69 YY_FLUSH_BUFFER

```
#define YY_FLUSH_BUFFER yy_flush_buffer( YY_CURRENT_BUFFER )
```

10.1.1.70 yy_init_buffer

```
#define yy_init_buffer annot_init_buffer
```


10.1.1.71 YY_INPUT

```
#define YY_INPUT(
    buf,
    result,
    max_size )
```

Value:

```
if ( YY_CURRENT_BUFFER_LVALUE->yy_is_interactive ) \
{ \
    int c = '*'; \
    int n; \
    for ( n = 0; n < max_size && \
          (c = getc( yyin )) != EOF && c != '\n'; ++n ) \
        buf[n] = (char) c; \
    if ( c == '\n' ) \
        buf[n++] = (char) c; \
    if ( c == EOF && ferror( yyin ) ) \
        YY_FATAL_ERROR( "input in flex scanner failed" ); \
    result = n; \
} \
else \
{ \
    errno=0; \
    while ( (result = (int) fread(buf, 1, (yy_size_t) max_size, yyin)) == 0 && ferror(yyin)) \
    { \
        if( errno != EINTR ) \
        { \
            YY_FATAL_ERROR( "input in flex scanner failed" ); \
            break; \
        } \
        errno=0; \
        clearerr(yyin); \
    } \
} \
\
```

10.1.1.72 YY_INT_ALIGNED

```
#define YY_INT_ALIGNED short int
```

10.1.1.73 YY_LESS_LINENO

```
#define YY_LESS_LINENO(
    n )
```

10.1.1.74 YY_LINENO_REWIND_TO

```
#define YY_LINENO_REWIND_TO(
    ptr )
```

10.1.1.75 yy_load_buffer_state

```
static void yy_load_buffer_state(  
    void ) annot_load_buffer_state
```

10.1.1.76 YY_MORE_ADJ

```
#define YY_MORE_ADJ 0
```

10.1.1.77 yy_new_buffer

```
#define yy_new_buffer yy_create_buffer
```

10.1.1.78 YY_NEW_FILE

```
#define YY_NEW_FILE yyrestart( yyin )
```

10.1.1.79 YY_NO_INPUT

```
#define YY_NO_INPUT 1
```

10.1.1.80 YY_NULL

```
#define YY_NULL 0
```

10.1.1.81 YY_NUM_RULES

```
#define YY_NUM_RULES 15
```

10.1.1.82 YY_READ_BUF_SIZE

```
#define YY_READ_BUF_SIZE 8192
```

10.1.1.83 YY_RESTORE_YY_MORE_OFFSET

```
#define YY_RESTORE_YY_MORE_OFFSET
```

10.1.1.84 YY_RULE_SETUP

```
#define YY_RULE_SETUP YY_USER_ACTION
```

10.1.1.85 YY_SC_TO_UI

```
#define YY_SC_TO_UI(  
    c ) ((YY_CHAR) (c))
```

10.1.1.86 yy_scan_buffer

```
#define yy_scan_buffer annot_scan_buffer
```

10.1.1.87 yy_scan_bytes

```
#define yy_scan_bytes annot_scan_bytes
```

10.1.1.88 yy_scan_string

```
#define yy_scan_string annot_scan_string
```

10.1.1.89 yy_set_bol

```
#define yy_set_bol(  
    at_bol )
```

Value:

```
{ \n  
if ( ! YY_CURRENT_BUFFER ){ \n  
    yyensure_buffer_stack (); \n  
    YY_CURRENT_BUFFER_LVALUE = \n        yy_create_buffer( yyin, YY_BUF_SIZE ); \n  
} \n  
YY_CURRENT_BUFFER_LVALUE->yy_at_bol = at_bol; \n  
}
```

10.1.1.90 yy_set_interactive

```
#define yy_set_interactive(  
    is_interactive )
```

Value:

```
{ \n  
if ( ! YY_CURRENT_BUFFER ){ \n  
    yyensure_buffer_stack (); \n  
    YY_CURRENT_BUFFER_LVALUE = \n        yy_create_buffer( yyin, YY_BUF_SIZE ); \n  
} \n  
YY_CURRENT_BUFFER_LVALUE->yy_is_interactive = is_interactive; \n  
}
```

10.1.1.91 YY_SKIP_YYWRAP

```
#define YY_SKIP_YYWRAP
```

10.1.1.92 YY_START

```
#define YY_START (((yy_start) - 1) / 2)
```

10.1.1.93 YY_START_STACK_INCR

```
#define YY_START_STACK_INCR 25
```

10.1.1.94 YY_STATE_BUF_SIZE

```
#define YY_STATE_BUF_SIZE ((YY_BUF_SIZE + 2) * sizeof(yy_state_type))
```

10.1.1.95 YY_STATE_EOF

```
#define YY_STATE_EOF(  
    state ) (YY_END_OF_BUFFER + state + 1)
```

10.1.1.96 YY_STRUCT_YY_BUFFER_STATE

```
#define YY_STRUCT_YY_BUFFER_STATE
```

10.1.1.97 yy_switch_to_buffer

```
#define yy_switch_to_buffer annot_switch_to_buffer
```

10.1.1.98 YY_TYPEDEF_YY_BUFFER_STATE

```
#define YY_TYPEDEF_YY_BUFFER_STATE
```

10.1.1.99 YY_TYPEDEF_YY_SIZE_T

```
#define YY_TYPEDEF_YY_SIZE_T
```

10.1.1.100 YY_USER_ACTION

```
#define YY_USER_ACTION aloc.columns (yylen);
```

10.1.1.101 yyalloc

```
#define yyalloc annotalloc
```

10.1.1.102 yyconst

```
#define yyconst const
```

10.1.1.103 yyensure_buffer_stack

```
static void yyensure_buffer_stack(  
    void ) annotensure_buffer_stack
```

10.1.1.104 yyfree

```
#define yyfree annotfree
```

10.1.1.105 yyget_debug

```
int yyget_debug(  
    void ) annotget_debug
```

10.1.1.106 yyget_extra

```
YY_EXTRA_TYPE yyget_extra annotget_extra
```

10.1.1.107 yyget_in

```
FILE * yyget_in(  
    void ) annotget_in
```

Get the input stream.

10.1.1.108 yyget_leng

```
int yyget_leng(  
    void ) annotget_leng
```

Get the length of the current token.

10.1.1.109 yyget_lineno

```
int yyget_lineno(  
    void ) annotget_lineno
```

Get the current line number.

10.1.1.110 yyget_out

```
FILE * yyget_out(  
    void ) annotget_out
```

Get the output stream.

10.1.1.111 yyget_text

```
char * yyget_text(  
    void ) annotget_text
```

Get the current token.

10.1.1.112 yyin

```
FILE * yyin annotin
```

10.1.1.113 yyleng

```
int yyleng annotleng
```

10.1.1.114 yyleless [1/2]

```
#define yyleless(
    n )
```

Value:

```
do \
{ \
    /* Undo effects of setting up yytext. */ \
    int yyless_macro_arg = (n); \
    YY_LESS_LINENO(yyless_macro_arg);\
    *yy_cp = (yy_hold_char); \
    YY_RESTORE_YY_MORE_OFFSET \
    (yy_c_buf_p) = yy_cp = yy_bp + yyless_macro_arg - YY_MORE_ADJ; \
    YY_DO_BEFORE_ACTION; /* set up yytext again */ \
} \
while ( 0 )
```

10.1.1.115 yyleless [2/2]

```
#define yyleless(
    n )
```

Value:

```
do \
{ \
    /* Undo effects of setting up yytext. */ \
    int yyless_macro_arg = (n); \
    YY_LESS_LINENO(yyless_macro_arg);\
    yytext[yy_leng] = (yy_hold_char); \
    (yy_c_buf_p) = yytext + yyless_macro_arg; \
    (yy_hold_char) = *(yy_c_buf_p); \
    *(yy_c_buf_p) = '\0'; \
    yy_leng = yyless_macro_arg; \
} \
while ( 0 )
```

10.1.1.116 yylex

```
int yylex annotlex
```

10.1.1.117 yylex_destroy

```
int yylex_destroy(
    void ) annotlex_destroy
```

10.1.1.118 yylex_init

```
#define yylex_init annotlex_init
```


10.1.1.119 yylex_init_extra

```
#define yylex_init_extra annotlex_init_extra
```

10.1.1.120 yylineno

```
int yylineno annotlineno
```

10.1.1.121 yymore

```
#define yymore( ) yymore_used_but_not_detected
```

10.1.1.122 yynoreturn

```
#define yynoreturn
```

10.1.1.123 yyout

```
FILE * yyout annotout
```

10.1.1.124 yypop_buffer_state

```
void yypop_buffer_state(  
    void ) annotpop_buffer_state
```

Removes and deletes the top of the stack, if present. The next element becomes the new top.

10.1.1.125 yypush_buffer_state

```
#define yypush_buffer_state annotpush_buffer_state
```

10.1.1.126 yyrealloc

```
#define yyrealloc annotrealloc
```

10.1.1.127 yyrestart

```
#define yyrestart annotrestart
```

10.1.1.128 yyset_debug

```
#define yyset_debug annotset_debug
```

10.1.1.129 yyset_extra

```
#define yyset_extra annotset_extra
```

10.1.1.130 yyset_in

```
#define yyset_in annotset_in
```

10.1.1.131 yyset_lineno

```
#define yyset_lineno annotset_lineno
```

10.1.1.132 yyset_out

```
#define yyset_out annotset_out
```

10.1.1.133 YYSTATE

```
#define YYSTATE YY_START
```

10.1.1.134 YYTABLES_NAME

```
#define YYTABLES_NAME "yytables"
```

10.1.1.135 yyterminate

```
#define yyterminate( ) return YY_NULL
```

10.1.1.136 yytext

```
char * yytext annottext
```

10.1.1.137 yytext_ptr

```
#define yytext_ptr yytext
```

10.1.1.138 yywrap

```
#define yywrap annotwrap
```

10.1.2 Typedef Documentation

10.1.2.1 flex_int16_t

```
typedef short int flex_int16_t
```

10.1.2.2 flex_int32_t

```
typedef int flex_int32_t
```

10.1.2.3 flex_int8_t

```
typedef signed char flex_int8_t
```

10.1.2.4 flex_uint16_t

```
typedef unsigned short int flex_uint16_t
```

10.1.2.5 flex_uint32_t

```
typedef unsigned int flex_uint32_t
```

10.1.2.6 flex_uint8_t

```
typedef unsigned char flex_uint8_t
```

10.1.2.7 YY_BUFFER_STATE

```
typedef struct yy_buffer_state* YY_BUFFER_STATE
```

10.1.2.8 YY_CHAR

```
typedef flex_uint8_t YY_CHAR
```

10.1.2.9 yy_size_t

```
typedef size_t yy_size_t
```

10.1.2.10 yy_state_type

```
typedef int yy_state_type
```

10.1.3 Function Documentation

10.1.3.1 if()

```
if (
    ! yy_init )
```

10.1.3.2 step()

```
alloc step ( )
```

10.1.3.3 stripquotes()

```
string stripquotes (
    string s )
```

10.1.3.4 while()

```
while (
    1 )
```

10.1.3.5 yy_create_buffer()

```
YY_BUFFER_STATE yy_create_buffer (
    FILE * file,
    int size )
```

Allocate and initialize an input buffer state.

Parameters

<i>file</i>	A readable stream.
<i>size</i>	The character buffer size in bytes. When in doubt, use YY_BUF_SIZE.

Returns

the allocated buffer state.

10.1.3.6 yy_delete_buffer()

```
void yy_delete_buffer (
    YY_BUFFER_STATE b )
```

Destroy the buffer.

Parameters

<i>b</i>	a buffer created with yy_create_buffer()
----------	--

10.1.3.7 yy_flush_buffer()

```
void yy_flush_buffer (
    YY_BUFFER_STATE b )
```

Discard all buffered characters. On the next scan, YY_INPUT will be called.

Parameters

<i>b</i>	the buffer state to be flushed, usually YY_CURRENT_BUFFER.
----------	--

10.1.3.8 yy_scan_buffer()

```
YY_BUFFER_STATE yy_scan_buffer (
    char * base,
    yy_size_t size )
```

Setup the input buffer state to scan directly from a user-specified character buffer.

Parameters

<i>base</i>	the character buffer
<i>size</i>	the size in bytes of the character buffer

Returns

the newly allocated buffer state object.

10.1.3.9 yy_scan_bytes()

```
YY_BUFFER_STATE yy_scan_bytes (
    const char * yybytes,
    int _yybytes_len )
```

Setup the input buffer state to scan the given bytes. The next call to [yylex\(\)](#) will scan from a *copy* of *bytes*.

Parameters

<i>yybytes</i>	the byte buffer to scan
<i>_yybytes_len</i>	the number of bytes in the buffer pointed to by <i>bytes</i> .

Returns

the newly allocated buffer state object.

10.1.3.10 yy_scan_string()

```
YY_BUFFER_STATE yy_scan_string (
    const char * yystr )
```

Setup the input buffer state to scan a string. The next call to [yylex\(\)](#) will scan from a *copy* of *str*.

Parameters

<i>yystr</i>	a NUL-terminated string to scan
--------------	---------------------------------

Returns

the newly allocated buffer state object.

Note

If you want to scan bytes that may contain NUL values, then use [yy_scan_bytes\(\)](#) instead.

10.1.3.11 yy_switch_to_buffer()

```
void yy_switch_to_buffer (
    YY_BUFFER_STATE new_buffer )
```

Switch to a different input buffer.

Parameters

<i>new_buffer</i>	The new input buffer.
-------------------	-----------------------

10.1.3.12 yyalloc()

```
void * yyalloc (
    yy_size_t size )
```

10.1.3.13 yyfree()

```
void yyfree (
    void * ptr )
```

10.1.3.14 yypush_buffer_state()

```
void yypush_buffer_state (
    YY_BUFFER_STATE new_buffer )
```

Pushes the new state onto the stack. The new state becomes the current state. This function will allocate the stack if necessary.

Parameters

<i>new_buffer</i>	The new state.
-------------------	----------------

10.1.3.15 yyrealloc()

```
void * yyrealloc (
    void * ptr,
    yy_size_t size )
```

10.1.3.16 yyrestart()

```
void yyrestart (
    FILE * input_file )
```

Immediately switch to a different input stream.

Parameters

<code>input_file</code>	A readable stream.
-------------------------	--------------------

Note

This function does not reset the start condition to `INITIAL`.

10.1.3.17 yyset_debug()

```
void yyset_debug (
    int debug_flag )
```

10.1.3.18 yyset_extra()

```
void yyset_extra (
    YY_EXTRA_TYPE user_defined )
```

10.1.3.19 yyset_in()

```
void yyset_in (
    FILE * _in_str )
```

Set the input stream. This does not discard the current input buffer.

Parameters

<code>_in_str</code>	A readable stream.
----------------------	--------------------

See also

[yy_switch_to_buffer](#)

10.1.3.20 yyset_lineno()

```
void yyset_lineno (
    int _line_number )
```

Set the current line number.

Parameters

<code>_line_number</code>	line number
---------------------------	-------------

10.1.3.21 yyset_out()

```
void yyset_out (
    FILE * _out_str )
```

10.1.4 Variable Documentation**10.1.4.1 yy_act**

```
int yy_act
```

10.1.4.2 yy_bp

```
char * yy_bp
```

10.1.4.3 yy_cp

```
char* yy_cp
```

10.1.4.4 YY_DECL

```
YY_DECL
```

Initial value:

```
{
    yy_state_type yy_current_state
```

The main scanner function which does all the work.

10.1.4.5 yy_flex_debug

```
int yy_flex_debug = 0
```

10.1.4.6 yyin

```
FILE* yyin = NULL
```

10.1.4.7 yyleng

```
int yyleng
```

10.1.4.8 yylineno

```
int yylineno = 1
```

10.1.4.9 yyout

```
FILE * yyout = NULL
```

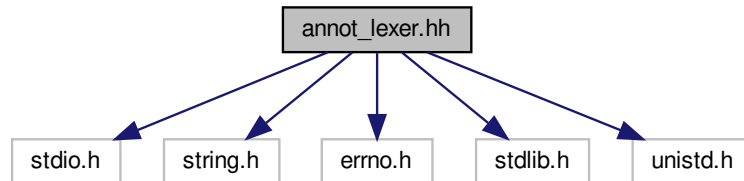
10.1.4.10 yytext

```
char* yytext
```

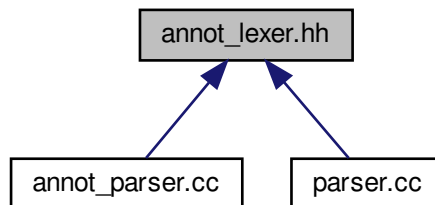
10.2 annot_lexer.hh File Reference

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
```

Include dependency graph for annot_lexer.hh:



This graph shows which files directly or indirectly include this file:



Classes

- struct [yy_buffer_state](#)

Macros

- #define [annotIN_HEADER](#) 1
- #define [YY_INT_ALIGNED](#) short int
- #define [FLEX_SCANNER](#)
- #define [YY_FLEX_MAJOR_VERSION](#) 2
- #define [YY_FLEX_MINOR_VERSION](#) 6
- #define [YY_FLEX_SUBMINOR_VERSION](#) 4
- #define [FLEX_BETA](#)
- #define [yy_create_buffer](#) annot_create_buffer

- `#define yy_delete_buffer` `annot_delete_buffer`
- `#define yy_scan_buffer` `annot_scan_buffer`
- `#define yy_scan_string` `annot_scan_string`
- `#define yy_scan_bytes` `annot_scan_bytes`
- `#define yy_init_buffer` `annot_init_buffer`
- `#define yy_flush_buffer` `annot_flush_buffer`
- `#define yy_load_buffer_state` `annot_load_buffer_state`
- `#define yy_switch_to_buffer` `annot_switch_to_buffer`
- `#define yypush_buffer_state` `annotpush_buffer_state`
- `#define yypop_buffer_state` `annotpop_buffer_state`
- `#define yyensure_buffer_stack` `annotensure_buffer_stack`
- `#define yylex` `annotlex`
- `#define yyrestart` `annotrestart`
- `#define yylex_init` `annotlex_init`
- `#define yylex_init_extra` `annotlex_init_extra`
- `#define yylex_destroy` `annotlex_destroy`
- `#define yyget_debug` `annotget_debug`
- `#define yyset_debug` `annotset_debug`
- `#define yyget_extra` `annotget_extra`
- `#define yyset_extra` `annotset_extra`
- `#define yyget_in` `annotget_in`
- `#define yyset_in` `annotset_in`
- `#define yyget_out` `annotget_out`
- `#define yyset_out` `annotset_out`
- `#define yyget_leng` `annotget_leng`
- `#define yyget_text` `annotget_text`
- `#define yyget_lineno` `annotget_lineno`
- `#define yyset_lineno` `annotset_lineno`
- `#define yywrap` `annotwrap`
- `#define yyalloc` `annotalloc`
- `#define yyrealloc` `annotrealloc`
- `#define yyfree` `annotfree`
- `#define yytext` `annottext`
- `#define yyleng` `annotleng`
- `#define yyin` `annotin`
- `#define yyout` `annotout`
- `#define yy_flex_debug` `annot_flex_debug`
- `#define yylineno` `annotlineno`
- `#define FLEXINT_H`
- `#define INT8_MIN` `(-128)`
- `#define INT16_MIN` `(-32767-1)`
- `#define INT32_MIN` `(-2147483647-1)`
- `#define INT8_MAX` `(127)`
- `#define INT16_MAX` `(32767)`
- `#define INT32_MAX` `(2147483647)`
- `#define UINT8_MAX` `(255U)`
- `#define UINT16_MAX` `(65535U)`
- `#define UINT32_MAX` `(4294967295U)`
- `#define SIZE_MAX` `(~(size_t)0)`
- `#define yyconst` `const`
- `#define yynoreturn`
- `#define YY_BUF_SIZE` `16384`
- `#define YY_TYPEDEF_YY_BUFFER_STATE`
- `#define YY_TYPEDEF_YY_SIZE_T`
- `#define YY_STRUCT_YY_BUFFER_STATE`

- `#define annotwrap()` (`/*CONSTCOND*/1`)
- `#define YY_SKIP_YYWRAP`
- `#define yytext_ptr yytext`
- `#define YY_EXTRA_TYPE void *`
- `#define YY_READ_BUF_SIZE 8192`
- `#define YY_START_STACK_INCR 25`
- `#define YY_DECL_IS_OURS 1`
- `#define YY_DECL int yylex (void)`

Typedefs

- `typedef signed char flex_int8_t`
- `typedef short int flex_int16_t`
- `typedef int flex_int32_t`
- `typedef unsigned char flex_uint8_t`
- `typedef unsigned short int flex_uint16_t`
- `typedef unsigned int flex_uint32_t`
- `typedef struct yy_buffer_state * YY_BUFFER_STATE`
- `typedef size_t yy_size_t`

Functions

- `void yyrestart (FILE *input_file)`
- `void yy_switch_to_buffer (YY_BUFFER_STATE new_buffer)`
- `YY_BUFFER_STATE yy_create_buffer (FILE *file, int size)`
- `void yy_delete_buffer (YY_BUFFER_STATE b)`
- `void yy_flush_buffer (YY_BUFFER_STATE b)`
- `void yypush_buffer_state (YY_BUFFER_STATE new_buffer)`
- `YY_BUFFER_STATE yy_scan_buffer (char *base, yy_size_t size)`
- `YY_BUFFER_STATE yy_scan_string (const char *yy_str)`
- `YY_BUFFER_STATE yy_scan_bytes (const char *bytes, int len)`
- `void * yyalloc (yy_size_t)`
- `void * yyrealloc (void *, yy_size_t)`
- `void yyfree (void *)`
- `void yyset_debug (int debug_flag)`
- `void yyset_extra (YY_EXTRA_TYPE user_defined)`
- `void yyset_in (FILE *_in_str)`
- `void yyset_out (FILE *_out_str)`
- `void yyset_lineno (int _line_number)`

Variables

- `int yyleng`
- `FILE * yyin`
- `FILE * yyout`
- `int yylineno`
- `char * yytext`

10.2.1 Macro Definition Documentation

10.2.1.1 annotIN_HEADER

```
#define annotIN_HEADER 1
```

10.2.1.2 annotwrap

```
#define annotwrap( ) (/*CONSTCOND*/1)
```

10.2.1.3 FLEX_BETA

```
#define FLEX_BETA
```

10.2.1.4 FLEX_SCANNER

```
#define FLEX_SCANNER
```

10.2.1.5 FLEXINT_H

```
#define FLEXINT_H
```

10.2.1.6 INT16_MAX

```
#define INT16_MAX (32767)
```

10.2.1.7 INT16_MIN

```
#define INT16_MIN (-32767-1)
```

10.2.1.8 INT32_MAX

```
#define INT32_MAX (2147483647)
```

10.2.1.9 INT32_MIN

```
#define INT32_MIN (-2147483647-1)
```

10.2.1.10 INT8_MAX

```
#define INT8_MAX (127)
```

10.2.1.11 INT8_MIN

```
#define INT8_MIN (-128)
```

10.2.1.12 SIZE_MAX

```
#define SIZE_MAX (~(size_t)0)
```

10.2.1.13 UINT16_MAX

```
#define UINT16_MAX (65535U)
```

10.2.1.14 UINT32_MAX

```
#define UINT32_MAX (4294967295U)
```

10.2.1.15 UINT8_MAX

```
#define UINT8_MAX (255U)
```

10.2.1.16 YY_BUF_SIZE

```
#define YY_BUF_SIZE 16384
```


10.2.1.17 yy_create_buffer

```
#define yy_create_buffer annot_create_buffer
```

10.2.1.18 YY_DECL

```
#define YY_DECL int yylex (void)
```

10.2.1.19 YY_DECL_IS_OURS

```
#define YY_DECL_IS_OURS 1
```

10.2.1.20 yy_delete_buffer

```
#define yy_delete_buffer annot_delete_buffer
```

10.2.1.21 YY_EXTRA_TYPE

```
#define YY_EXTRA_TYPE void *
```

10.2.1.22 yy_flex_debug

```
#define yy_flex_debug annot_flex_debug
```

10.2.1.23 YY_FLEX_MAJOR_VERSION

```
#define YY_FLEX_MAJOR_VERSION 2
```

10.2.1.24 YY_FLEX_MINOR_VERSION

```
#define YY_FLEX_MINOR_VERSION 6
```

10.2.1.25 YY_FLEX_SUBMINOR_VERSION

```
#define YY_FLEX_SUBMINOR_VERSION 4
```

10.2.1.26 yy_flush_buffer

```
#define yy_flush_buffer annot_flush_buffer
```

10.2.1.27 yy_init_buffer

```
#define yy_init_buffer annot_init_buffer
```

10.2.1.28 YY_INT_ALIGNED

```
#define YY_INT_ALIGNED short int
```

10.2.1.29 yy_load_buffer_state

```
#define yy_load_buffer_state(  
    void ) annot_load_buffer_state
```

10.2.1.30 YY_READ_BUF_SIZE

```
#define YY_READ_BUF_SIZE 8192
```

10.2.1.31 yy_scan_buffer

```
#define yy_scan_buffer annot_scan_buffer
```

10.2.1.32 yy_scan_bytes

```
#define yy_scan_bytes annot_scan_bytes
```

10.2.1.33 yy_scan_string

```
#define yy_scan_string annot_scan_string
```

10.2.1.34 YY_SKIP_YYWRAP

```
#define YY_SKIP_YYWRAP
```

10.2.1.35 YY_START_STACK_INCR

```
#define YY_START_STACK_INCR 25
```

10.2.1.36 YY_STRUCT_YY_BUFFER_STATE

```
#define YY_STRUCT_YY_BUFFER_STATE
```

10.2.1.37 yy_switch_to_buffer

```
#define yy_switch_to_buffer annot_switch_to_buffer
```

10.2.1.38 YY_TYPEDEF_YY_BUFFER_STATE

```
#define YY_TYPEDEF_YY_BUFFER_STATE
```

10.2.1.39 YY_TYPEDEF_YY_SIZE_T

```
#define YY_TYPEDEF_YY_SIZE_T
```

10.2.1.40 yyalloc

```
#define yyalloc annotalloc
```

10.2.1.41 yyconst

```
#define yyconst const
```

10.2.1.42 yyensure_buffer_stack

```
#define yyensure_buffer_stack(  
    void ) annotensure_buffer_stack
```

10.2.1.43 yyfree

```
#define yyfree annotfree
```

10.2.1.44 yyget_debug

```
#define yyget_debug(  
    void ) annotget_debug
```

10.2.1.45 yyget_extra

```
#define yyget_extra(  
    void ) annotget_extra
```

10.2.1.46 yyget_in

```
#define yyget_in(  
    void ) annotget_in
```

10.2.1.47 yyget_leng

```
#define yyget_leng(  
    void ) annotget_leng
```

10.2.1.48 yyget_lineno

```
#define yyget_lineno(  
    void ) annotget_lineno
```

10.2.1.49 yyget_out

```
#define yyget_out(  
    void ) annotget_out
```

10.2.1.50 yyget_text

```
#define yyget_text(  
    void ) annotget_text
```

10.2.1.51 yyin

```
#define yyin annotin
```

10.2.1.52 yyleng

```
#define yyleng annotleng
```

10.2.1.53 yylex

```
#define yylex(  
    void ) annotlex
```

10.2.1.54 `yylex_destroy`

```
#define yylex_destroy(  
    void ) annotlex_destroy
```

10.2.1.55 `yylex_init`

```
#define yylex_init annotlex_init
```

10.2.1.56 `yylex_init_extra`

```
#define yylex_init_extra annotlex_init_extra
```

10.2.1.57 `yylineno`

```
#define yylineno annotlineno
```

10.2.1.58 `yynoreturn`

```
#define yynoreturn
```

10.2.1.59 `yyout`

```
#define yyout annotout
```

10.2.1.60 `yypop_buffer_state`

```
#define yypop_buffer_state(  
    void ) annotpop_buffer_state
```

10.2.1.61 yypush_buffer_state

```
#define yypush_buffer_state annotpush_buffer_state
```

10.2.1.62 yyrealloc

```
#define yyrealloc annotrealloc
```

10.2.1.63 yyrestart

```
#define yyrestart annotrestart
```

10.2.1.64 yyset_debug

```
#define yyset_debug annotset_debug
```

10.2.1.65 yyset_extra

```
#define yyset_extra annotset_extra
```

10.2.1.66 yyset_in

```
#define yyset_in annotset_in
```

10.2.1.67 yyset_lineno

```
#define yyset_lineno annotset_lineno
```

10.2.1.68 yyset_out

```
#define yyset_out annotset_out
```

10.2.1.69 yytext

```
#define yytext annottext
```

10.2.1.70 yytext_ptr

```
#define yytext_ptr yytext
```

10.2.1.71 yywrap

```
#define yywrap annotwrap
```

10.2.2 Typedef Documentation

10.2.2.1 flex_int16_t

```
typedef short int flex_int16_t
```

10.2.2.2 flex_int32_t

```
typedef int flex_int32_t
```

10.2.2.3 flex_int8_t

```
typedef signed char flex_int8_t
```

10.2.2.4 flex_uint16_t

```
typedef unsigned short int flex_uint16_t
```


10.2.2.5 flex_uint32_t

```
typedef unsigned int flex_uint32_t
```

10.2.2.6 flex_uint8_t

```
typedef unsigned char flex_uint8_t
```

10.2.2.7 YY_BUFFER_STATE

```
typedef struct yy_buffer_state* YY_BUFFER_STATE
```

10.2.2.8 yy_size_t

```
typedef size_t yy_size_t
```

10.2.3 Function Documentation

10.2.3.1 yy_create_buffer()

```
YY_BUFFER_STATE yy_create_buffer (
    FILE * file,
    int size )
```

Allocate and initialize an input buffer state.

Parameters

<i>file</i>	A readable stream.
<i>size</i>	The character buffer size in bytes. When in doubt, use YY_BUF_SIZE.

Returns

the allocated buffer state.

10.2.3.2 yy_delete_buffer()

```
void yy_delete_buffer (
    YY_BUFFER_STATE b )
```

Destroy the buffer.

Parameters

<i>b</i>	a buffer created with <code>yy_create_buffer()</code>
----------	---

10.2.3.3 yy_flush_buffer()

```
void yy_flush_buffer (
    YY_BUFFER_STATE b )
```

Discard all buffered characters. On the next scan, YY_INPUT will be called.

Parameters

<i>b</i>	the buffer state to be flushed, usually YY_CURRENT_BUFFER.
----------	--

10.2.3.4 yy_scan_buffer()

```
YY_BUFFER_STATE yy_scan_buffer (
    char * base,
    yy_size_t size )
```

Setup the input buffer state to scan directly from a user-specified character buffer.

Parameters

<i>base</i>	the character buffer
<i>size</i>	the size in bytes of the character buffer

Returns

the newly allocated buffer state object.

10.2.3.5 yy_scan_bytes()

```
YY_BUFFER_STATE yy_scan_bytes (
    const char * bytes,
    int len )
```

Setup the input buffer state to scan the given bytes. The next call to [yylex\(\)](#) will scan from a *copy* of *bytes*.

Parameters

<i>yybytes</i>	the byte buffer to scan
<i>_yybytes_len</i>	the number of bytes in the buffer pointed to by <i>bytes</i> .

Returns

the newly allocated buffer state object.

10.2.3.6 yy_scan_string()

```
YY_BUFFER_STATE yy_scan_string (  
    const char * yy_str )
```

Setup the input buffer state to scan a string. The next call to [yylex\(\)](#) will scan from a *copy* of *str*.

Parameters

<i>yyst</i>	a NUL-terminated string to scan
-------------	---------------------------------

Returns

the newly allocated buffer state object.

Note

If you want to scan bytes that may contain NUL values, then use [yy_scan_bytes\(\)](#) instead.

10.2.3.7 yy_switch_to_buffer()

```
void yy_switch_to_buffer (  
    YY_BUFFER_STATE new_buffer )
```

Switch to a different input buffer.

Parameters

<i>new_buffer</i>	The new input buffer.
-------------------	-----------------------

10.2.3.8 yyallocc()

```
void* yyallocc (
    yy_size_t size )
```

10.2.3.9 yyfree()

```
void yyfree (
    void * ptr )
```

10.2.3.10 yypush_buffer_state()

```
void yypush_buffer_state (
    YY_BUFFER_STATE new_buffer )
```

Pushes the new state onto the stack. The new state becomes the current state. This function will allocate the stack if necessary.

Parameters

<i>new_buffer</i>	The new state.
-------------------	----------------

10.2.3.11 yyrealloc()

```
void* yyrealloc (
    void * ptr,
    yy_size_t size )
```

10.2.3.12 yyrestart()

```
void yyrestart (
    FILE * input_file )
```

Immediately switch to a different input stream.

Parameters

<i>input_file</i>	A readable stream.
-------------------	--------------------

Note

This function does not reset the start condition to `INITIAL`.

10.2.3.13 yyset_debug()

```
void yyset_debug (
    int debug_flag )
```

10.2.3.14 yyset_extra()

```
void yyset_extra (
    YY_EXTRA_TYPE user_defined )
```

10.2.3.15 yyset_in()

```
void yyset_in (
    FILE * _in_str )
```

Set the input stream. This does not discard the current input buffer.

Parameters

<code>_in_str</code>	A readable stream.
----------------------	--------------------

See also

[yy_switch_to_buffer](#)

10.2.3.16 yyset_lineno()

```
void yyset_lineno (
    int _line_number )
```

Set the current line number.

Parameters

<code>_line_number</code>	line number
---------------------------	-------------

10.2.3.17 yyset_out()

```
void yyset_out (
    FILE * _out_str )
```

10.2.4 Variable Documentation

10.2.4.1 yyin

```
FILE* yyin [extern]
```

10.2.4.2 yyleng

```
int yyleng [extern]
```

10.2.4.3 yylineno

```
int yylineno [extern]
```

10.2.4.4 yyout

```
FILE * yyout
```

10.2.4.5 yytext

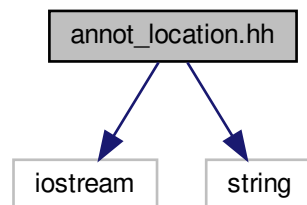
```
char* yytext [extern]
```

10.3 annot_location.hh File Reference

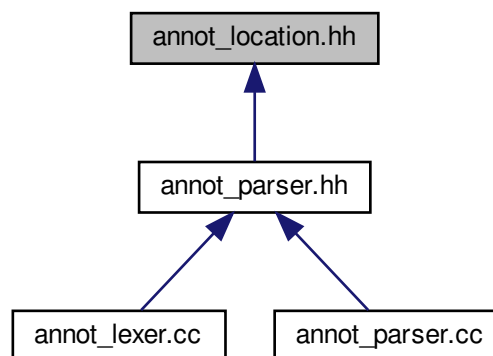
```
#include <iostream>
```

```
#include <string>
```

Include dependency graph for annot_location.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class `annot::position`
A point in a source file.
- class `annot::location`
Two points in a source file.

Namespaces

- `annot`

Macros

- `#define YY_NULLPTR ((void*)0)`

Functions

- `position & annot::operator+= (position &res, position::counter_type width)`
Add width columns, in place.
- `position annot::operator+ (position res, position::counter_type width)`
Add width columns.
- `position & annot::operator-= (position &res, position::counter_type width)`
Subtract width columns, in place.
- `position annot::operator- (position res, position::counter_type width)`
Subtract width columns.
- `template<typename YYChar >
std::basic_ostream< YYChar > & annot::operator<< (std::basic_ostream< YYChar > &ostr, const position
&pos)`
Intercept output stream redirection.
- `location & annot::operator+= (location &res, const location &end)`
Join two locations, in place.
- `location annot::operator+ (location res, const location &end)`
Join two locations.
- `location & annot::operator+= (location &res, location::counter_type width)`
Add width columns to the end position, in place.
- `location annot::operator+ (location res, location::counter_type width)`
Add width columns to the end position.
- `location & annot::operator-= (location &res, location::counter_type width)`
Subtract width columns to the end position, in place.
- `location annot::operator- (location res, location::counter_type width)`
Subtract width columns to the end position.
- `template<typename YYChar >
std::basic_ostream< YYChar > & annot::operator<< (std::basic_ostream< YYChar > &ostr, const location
&loc)`
Intercept output stream redirection.

10.3.1 Detailed Description

Define the `annot::location` class.

10.3.2 Macro Definition Documentation

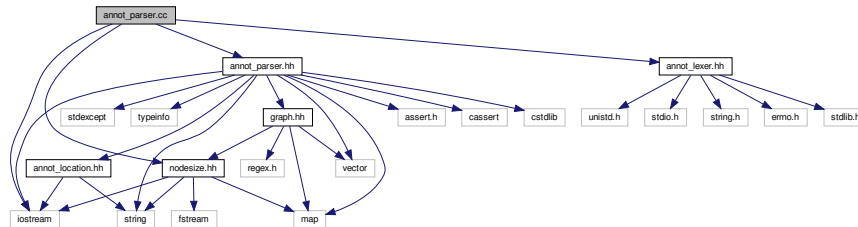
10.3.2.1 YY_NULLPTR

```
#define YY_NULLPTR ((void*)0)
```


10.4 annot_parser.cc File Reference

```
#include "annot_parser.hh"
#include <iostream>
#include "annot_lexer.hh"
#include "nodesize.hh"
```

Include dependency graph for annot_parser.cc:



Namespaces

- [annot](#)

Macros

- `#define yylex annotlex`
- `#define YY_DECL annot::parser::symbol_type annotlex (annotmap *m)`
- `#define YY_\(msgid\) msgid`
- `#define YY_EXCEPTIONS 1`
- `#define YYRHSLOC(Rhs, K) ((Rhs)[K].location)`
- `#define YYLLOC_DEFAULT(Current, Rhs, N)`
- `#define YYCDEBUG if (false) std::cerr`
- `#define YY_SYMBOL_PRINT(Title, Symbol) YY_USE (Symbol)`
- `#define YY_REDUCE_PRINT(Rule) static_cast<void> (0)`
- `#define YY_STACK_PRINT() static_cast<void> (0)`
- `#define yyerrok (yyerrstatus_ = 0)`
- `#define yyclearin (yyla.clear ())`
- `#define YYACCEPT goto yyacceptlab`
- `#define YYABORT goto yyabortlab`
- `#define YYERROR goto yyerrorlab`
- `#define YYRECOVERING() (!yyerrstatus_)`
- `#define YYCASE_(N, S)`

Functions

- [annotmap](#) * [scanAnnot](#) (string &s, void *loc)

Variables

- [annot::location](#) aloc
- [YY_DECL](#)
- [annot::location](#) loc

10.4.1 Macro Definition Documentation

10.4.1.1 YY_

```
#define YY_(  
    msgid ) msgid
```

10.4.1.2 YY_DECL

```
#define YY_DECL annot::parser::symbol\_type annotlex ( annotmap *m)
```

10.4.1.3 YY_EXCEPTIONS

```
#define YY_EXCEPTIONS 1
```

10.4.1.4 YY_REDUCE_PRINT

```
#define YY_REDUCE_PRINT(  
    Rule ) static_cast<void> (0)
```

10.4.1.5 YY_STACK_PRINT

```
#define YY_STACK_PRINT( ) static_cast<void> (0)
```

10.4.1.6 YY_SYMBOL_PRINT

```
#define YY_SYMBOL_PRINT(  
    Title,  
    Symbol ) YY\_USE (Symbol)
```

```
#define YYABORT goto yyabortlab
```

```
#define YYACCEPT goto yyacceptlab
```

```
#define YYCASE_(  
    N,  
    S )
```

```
case N:
    yyformat = S;
break
```

/ /

```
#define YYCDEBUG if (false) std::cerr
```

```
#define yyclearin (yyla.clear ())
```

```
#define yyerrok (yyerrstatus_ = 0)
```

```
#define YYERROR goto yyerrorlab
```

10.4.1.14 yylex

```
#define yylex(
    void ) annotlex
```

10.4.1.15 YYLLOC_DEFAULT

```
#define YYLLOC_DEFAULT(
    Current,
    RhS,
    N )
```

Value:

```
do
  if (N)
  {
    (Current).begin = YYRHSLOC (RhS, 1).begin;
    (Current).end   = YYRHSLOC (RhS, N).end;
  }
  else
  {
    (Current).begin = (Current).end = YYRHSLOC (RhS, 0).end;
  }
  while (false)
```

10.4.1.16 YYRECOVERING

```
#define YYRECOVERING( ) (!yyerrstatus_)
```

10.4.1.17 YYRHSLOC

```
#define YYRHSLOC(
    RhS,
    K ) ((RhS)[K].location)
```

10.4.2 Function Documentation

10.4.2.1 scanAnnot()

```
annotmap* scanAnnot (
    string & s,
    void * loc )
```

10.4.3 Variable Documentation

10.4.3.1 aloc

```
annot::location aloc
```

10.4.3.2 loc

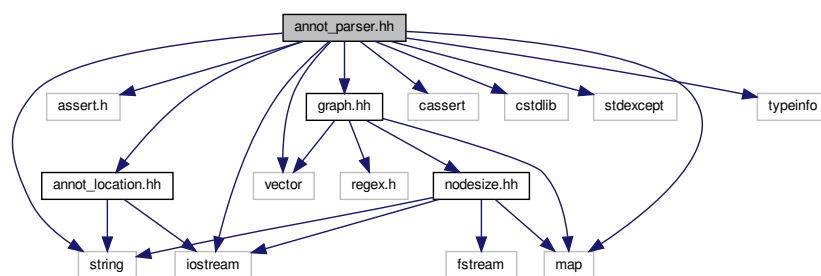
```
annot::location loc
```

10.4.3.3 YY_DECL

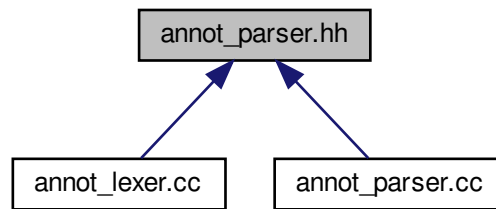
```
YY_DECL
```

10.5 annot_parser.hh File Reference

```
#include <string>
#include <assert.h>
#include <map>
#include <graph.hh>
#include <cassert>
#include <cstdlib>
#include <iostream>
#include <stdexcept>
#include <vector>
#include "annot_location.hh"
#include <typeinfo>
Include dependency graph for annot_parser.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [annot::parser](#)
A Bison parser.
- class [annot::parser::semantic_type](#)
- struct [annot::parser::syntax_error](#)
Syntax errors thrown from user actions.
- struct [annot::parser::token](#)
Token kinds.
- struct [annot::parser::symbol_kind](#)
Symbol kinds.
- struct [annot::parser::basic_symbol< Base >](#)
- struct [annot::parser::by_kind](#)
Type access provider for token (enum) based symbols.
- struct [annot::parser::symbol_type](#)
"External" symbols: returned by the scanner.
- class [annot::parser::context](#)
- class [annot::parser::stack< T, S >::slice](#)
Present a slice of the top of a stack.

Namespaces

- [annot](#)

Macros

- `#define YY_CPLUSPLUS 199711L`
- `#define YY_MOVE`
- `#define YY_MOVE_OR_COPY copy`
- `#define YY_MOVE_REF(Type) Type&`
- `#define YY_RVREF(Type) const Type&`
- `#define YY_COPY(Type) const Type&`
- `#define YY_NOEXCEPT`
- `#define YY_NOTHROW throw ()`

- `#define YY_CONSTEXPR`
- `#define ANNOT_ASSERT assert`
- `#define YY_ATTRIBUTE_PURE`
- `#define YY_ATTRIBUTE_UNUSED`
- `#define YY_USE(E) ((void) (E))`
- `#define YY_INITIAL_VALUE(Value) Value`
- `#define YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN`
- `#define YY_IGNORE_MAYBE_UNINITIALIZED_END`
- `#define YY_IGNORE_USELESS_CAST_BEGIN`
- `#define YY_IGNORE_USELESS_CAST_END`
- `#define YY_CAST(Type, Val) ((Type) (Val))`
- `#define YY_REINTERPRET_CAST(Type, Val) ((Type) (Val))`
- `#define ANNOTDEBUG 0`

10.5.1 Detailed Description

Define the `annot::parser` class.

10.5.2 Macro Definition Documentation

10.5.2.1 ANNOT_ASSERT

```
#define ANNOT_ASSERT assert
```

10.5.2.2 ANNOTDEBUG

```
#define ANNOTDEBUG 0
```

10.5.2.3 YY_ATTRIBUTE_PURE

```
#define YY_ATTRIBUTE_PURE
```

10.5.2.4 YY_ATTRIBUTE_UNUSED

```
#define YY_ATTRIBUTE_UNUSED
```

10.5.2.5 YY_CAST

```
#define YY_CAST(  
    Type,  
    Val ) ((Type) (Val))
```

10.5.2.6 YY_CONSTEXPR

```
#define YY_CONSTEXPR
```

10.5.2.7 YY_COPY

```
#define YY_COPY(  
    Type ) const Type&
```

10.5.2.8 YY_CPLUSPLUS

```
#define YY_CPLUSPLUS 199711L
```

10.5.2.9 YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN

```
#define YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN
```

10.5.2.10 YY_IGNORE_MAYBE_UNINITIALIZED_END

```
#define YY_IGNORE_MAYBE_UNINITIALIZED_END
```

10.5.2.11 YY_IGNORE_USELESS_CAST_BEGIN

```
#define YY_IGNORE_USELESS_CAST_BEGIN
```


10.5.2.12 YY_IGNORE_USELESS_CAST_END

```
#define YY_IGNORE_USELESS_CAST_END
```

10.5.2.13 YY_INITIAL_VALUE

```
#define YY_INITIAL_VALUE(  
    Value ) Value
```

10.5.2.14 YY_MOVE

```
#define YY_MOVE
```

10.5.2.15 YY_MOVE_OR_COPY

```
#define YY_MOVE_OR_COPY copy
```

10.5.2.16 YY_MOVE_REF

```
#define YY_MOVE_REF(  
    Type ) Type&
```

10.5.2.17 YY_NOEXCEPT

```
#define YY_NOEXCEPT
```

10.5.2.18 YY_NOTHROW

```
#define YY_NOTHROW throw ()
```

10.5.2.19 YY_REINTERPRET_CAST

```
#define YY_REINTERPRET_CAST(  
    Type,  
    Val ) ((Type) (Val))
```

10.5.2.20 YY_RVREF

```
#define YY_RVREF(  
    Type ) const Type&
```

10.5.2.21 YY_USE

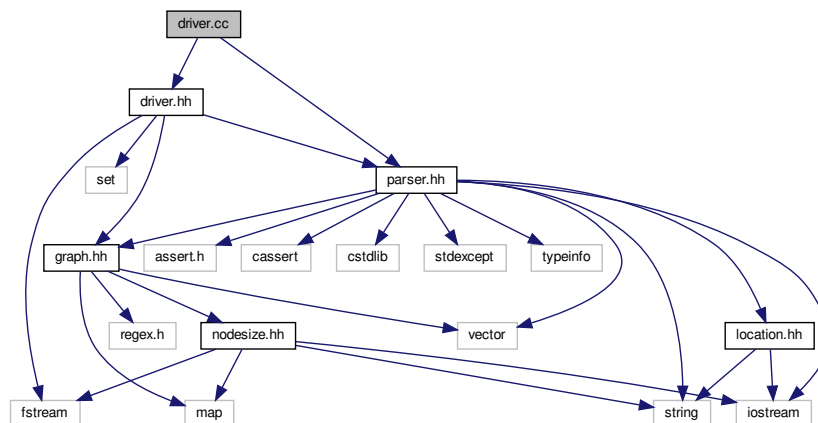
```
#define YY_USE(  
    E ) ((void) (E))
```

10.6 CODE_OF_CONDUCT.md File Reference

10.7 CONTRIBUTING.md File Reference

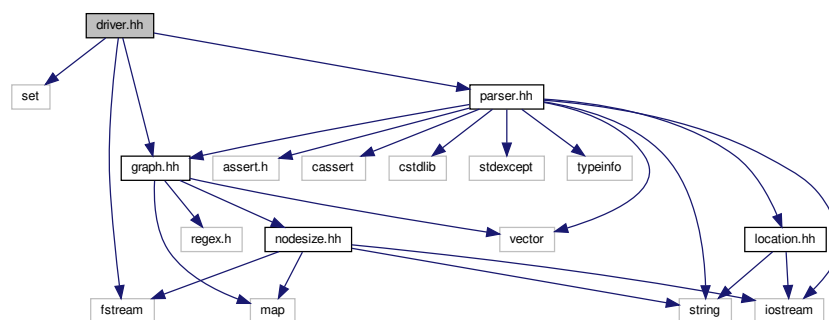
10.8 driver.cc File Reference

```
#include "driver.hh"  
#include "parser.hh"  
Include dependency graph for driver.cc:
```

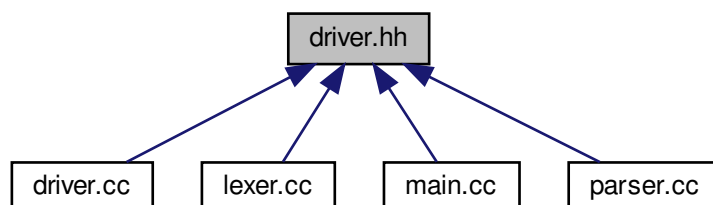


10.9 driver.hh File Reference

```
#include <set>
#include <fstream>
#include "graph.hh"
#include "parser.hh"
Include dependency graph for driver.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [driver](#)

Macros

- #define [YY_DECL yy::parser::symbol_type yylex \(driver& drv\)](#)

Variables

- [YY_DECL](#)

10.9.1 Macro Definition Documentation

10.9.1.1 YY_DECL

```
#define YY_DECL yy::parser::symbol_type yylex (driver& drv)
```

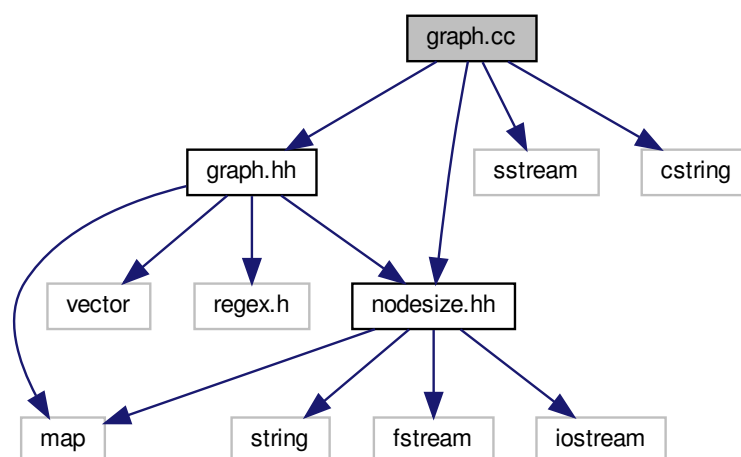
10.9.2 Variable Documentation

10.9.2.1 YY_DECL

```
YY_DECL
```

10.10 graph.cc File Reference

```
#include <graph.hh>
#include <sstream>
#include <nodesize.hh>
#include <cstring>
Include dependency graph for graph.cc:
```



Functions

- string [latexwrite](#) (string fontspec, string s)
- string [nextCoord](#) ()
- string [nextNode](#) ()
- string [nextChain](#) ()
- string [nextFit](#) ()
- string [stripSpecial](#) (string s)

10.10.1 Function Documentation

10.10.1.1 latexwrite()

```
string latexwrite (
    string fontspec,
    string s )
```

10.10.1.2 nextChain()

```
string nextChain ( )
```

10.10.1.3 nextCoord()

```
string nextCoord ( )
```

10.10.1.4 nextFit()

```
string nextFit ( )
```

10.10.1.5 nextNode()

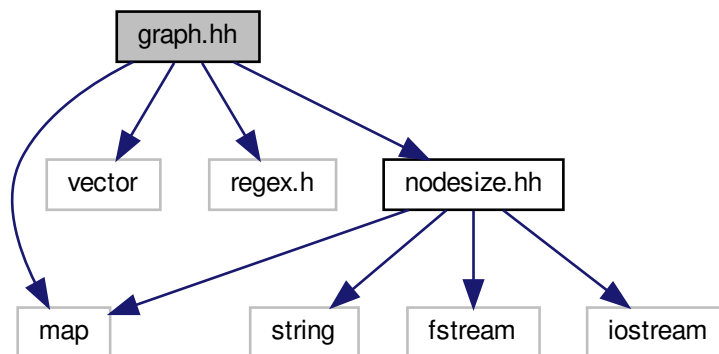
```
string nextNode ( )
```

10.10.1.6 stripSpecial()

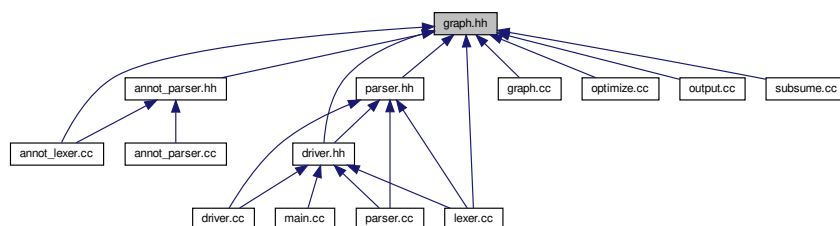
```
string stripSpecial (
    string s )
```

10.11 graph.hh File Reference

```
#include <map>
#include <vector>
#include <regex.h>
#include <nodesize.hh>
Include dependency graph for graph.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [node](#)
- class [singlenode](#)
- class [railnode](#)
- class [multinode](#)

- class [nontermnode](#)
- class [termnode](#)
- class [nullnode](#)
- class [newlinenode](#)
- class [rownode](#)
- class [choicenode](#)
- class [loopnode](#)
- class [concatnode](#)
- class [productionnode](#)
- class [grammar](#)

Typedefs

- typedef map< string, string > [annotmap](#)
- typedef pair< string, string > [annot_t](#)

Enumerations

- enum [vraillside](#) { [LEFT](#) , [RIGHT](#) }
- enum [vrailldir](#) { [UP](#) , [DOWN](#) }

Functions

- string [latexwrite](#) (string fontspec, string s)
- string [nextCoord](#) ()
- string [nextNode](#) ()
- string [nextChain](#) ()
- string [nextFit](#) ()
- string [stripSpecial](#) (string s)

10.11.1 Typedef Documentation

10.11.1.1 [annot_t](#)

```
typedef pair<string,string> annot\_t
```

10.11.1.2 [annotmap](#)

```
typedef map<string,string> annotmap
```

10.11.2 Enumeration Type Documentation

10.11.2.1 [vrailldir](#)

```
enum vrailldir
```

Enumerator

UP	
DOWN	

10.11.2.2 vrailside

```
enum vrailside
```

Enumerator

LEFT	
RIGHT	

10.11.3 Function Documentation**10.11.3.1 latexwrite()**

```
string latexwrite (
    string fontspec,
    string s )
```

10.11.3.2 nextChain()

```
string nextChain ( )
```

10.11.3.3 nextCoord()

```
string nextCoord ( )
```

10.11.3.4 nextFit()

```
string nextFit ( )
```


10.11.3.5 nextNode()

```
string nextNode ( )
```

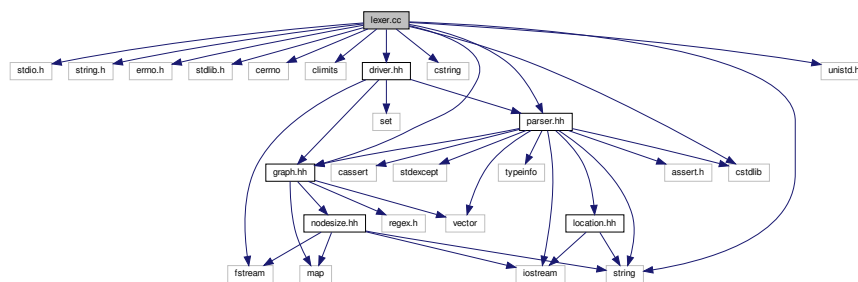
10.11.3.6 stripSpecial()

```
string stripSpecial (
    string s )
```

10.12 lexer.cc File Reference

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <cerrno>
#include <climits>
#include <cstdlib>
#include <cstring>
#include <string>
#include "graph.hh"
#include "driver.hh"
#include "parser.hh"
#include <unistd.h>
```

Include dependency graph for lexer.cc:



Classes

- struct [yy_buffer_state](#)
- struct [yy_trans_info](#)

Macros

- #define YY_INT_ALIGNED short int
- #define FLEX_SCANNER
- #define YY_FLEX_MAJOR_VERSION 2
- #define YY_FLEX_MINOR_VERSION 6
- #define YY_FLEX_SUBMINOR_VERSION 4
- #define FLEX_BETA
- #define FLEXINT_H
- #define INT8_MIN (-128)
- #define INT16_MIN (-32767-1)
- #define INT32_MIN (-2147483647-1)
- #define INT8_MAX (127)
- #define INT16_MAX (32767)
- #define INT32_MAX (2147483647)
- #define UINT8_MAX (255U)
- #define UINT16_MAX (65535U)
- #define UINT32_MAX (4294967295U)
- #define SIZE_MAX (~(size_t)0)
- #define yyconst const
- #define yynoreturn
- #define YY_NULL 0
- #define YY_SC_TO_UI(c) ((YY_CHAR) (c))
- #define BEGIN (yy_start) = 1 + 2 *
- #define YY_START (((yy_start) - 1) / 2)
- #define YYSTATE YY_START
- #define YY_STATE_EOF(state) (YY_END_OF_BUFFER + state + 1)
- #define YY_NEW_FILE yyrestart(yyin)
- #define YY_END_OF_BUFFER_CHAR 0
- #define YY_BUF_SIZE 16384
- #define YY_STATE_BUF_SIZE ((YY_BUF_SIZE + 2) * sizeof(yy_state_type))
- #define YY_TYPEDEF_YY_BUFFER_STATE
- #define YY_TYPEDEF_YY_SIZE_T
- #define EOB_ACT_CONTINUE_SCAN 0
- #define EOB_ACT_END_OF_FILE 1
- #define EOB_ACT_LAST_MATCH 2
- #define YY_LESS_LINENO(n)
- #define YY_LINENO_REWIND_TO(ptr)
- #define yless(n)
- #define unput(c) yyunput(c, (yytext_ptr))
- #define YY_STRUCT_YY_BUFFER_STATE
- #define YY_BUFFER_NEW 0
- #define YY_BUFFER_NORMAL 1
- #define YY_BUFFER_EOF_PENDING 2
- #define YY_CURRENT_BUFFER
- #define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]
- #define YY_FLUSH_BUFFER yy_flush_buffer(YY_CURRENT_BUFFER)
- #define yy_new_buffer yy_create_buffer
- #define yy_set_interactive(is_interactive)
- #define yy_set_bol(at_bol)
- #define YY_AT_BOL() (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)
- #define yywrap() (/*CONSTCOND*/1)
- #define YY_SKIP_YYWRAP
- #define FLEX_DEBUG
- #define yytext_ptr yytext

- `#define YY_DO_BEFORE_ACTION`
- `#define YY_NUM_RULES 27`
- `#define YY_END_OF_BUFFER 28`
- `#define REJECT reject_used_but_not_detected`
- `#define yymore() ((yy_more_flag) = 1)`
- `#define YY_MORE_ADJ (yy_more_len)`
- `#define YY_RESTORE_YY_MORE_OFFSET`
- `#define YY_NO_INPUT 1`
- `#define YY_USER_ACTION`
- `#define INITIAL 0`
- `#define A 1`
- `#define YY_EXTRA_TYPE void *`
- `#define YY_READ_BUF_SIZE 8192`
- `#define ECHO do { if (fwrite(yytext, (size_t) yyleng, 1, yyout)) {} } while (0)`
- `#define YY_INPUT(buf, result, max_size)`
- `#define yyterminate() return YY_NULL`
- `#define YY_START_STACK_INCR 25`
- `#define YY_FATAL_ERROR(msg) yy_fatal_error(msg)`
- `#define YY_BREAK /*LINTED*/break;`
- `#define YY_RULE_SETUP YY_USER_ACTION`
- `#define YY_EXIT_FAILURE 2`
- `#define yyless(n)`
- `#define YYTABLES_NAME "yytables"`

Typedefs

- `typedef signed char flex_int8_t`
- `typedef short int flex_int16_t`
- `typedef int flex_int32_t`
- `typedef unsigned char flex_uint8_t`
- `typedef unsigned short int flex_uint16_t`
- `typedef unsigned int flex_uint32_t`
- `typedef struct yy_buffer_state * YY_BUFFER_STATE`
- `typedef size_t yy_size_t`
- `typedef flex_uint8_t YY_CHAR`
- `typedef int yy_state_type`

Functions

- `void yyrestart (FILE *input_file)`
- `void yy_switch_to_buffer (YY_BUFFER_STATE new_buffer)`
- `YY_BUFFER_STATE yy_create_buffer (FILE *file, int size)`
- `void yy_delete_buffer (YY_BUFFER_STATE b)`
- `void yy_flush_buffer (YY_BUFFER_STATE b)`
- `void yypush_buffer_state (YY_BUFFER_STATE new_buffer)`
- `void yypop_buffer_state (void)`
- `YY_BUFFER_STATE yy_scan_buffer (char *base, yy_size_t size)`
- `YY_BUFFER_STATE yy_scan_string (const char *yy_str)`
- `YY_BUFFER_STATE yy_scan_bytes (const char *bytes, int len)`
- `void * yyalloc (yy_size_t)`
- `void * yyrealloc (void *, yy_size_t)`
- `void yyfree (void *)`
- `int yylex_destroy (void)`

- int `yyget_debug` (void)
- void `yyset_debug` (int debug_flag)
- `YY_EXTRA_TYPE` `yyget_extra` (void)
- void `yyset_extra` (`YY_EXTRA_TYPE` user_defined)
- FILE * `yyget_in` (void)
- void `yyset_in` (FILE * _in_str)
- FILE * `yyget_out` (void)
- void `yyset_out` (FILE * _out_str)
- int `yyget_leng` (void)
- char * `yyget_text` (void)
- int `yyget_lineno` (void)
- void `yyset_lineno` (int _line_number)
- `if` (!yy_init))
- `loc step` ()
- `while` (1)

Variables

- int `yyleng`
- FILE * `yyin` = NULL
- FILE * `yyout` = NULL
- int `yylineno` = 1
- char * `yytext`
- int `yy_flex_debug` = 1
- `YY_DECL`
- char * `yy_cp`
- char * `yy_bp`
- int `yy_act`
- `yy::location` subloc

10.12.1 Macro Definition Documentation

10.12.1.1 A

```
#define A 1
```

10.12.1.2 BEGIN

```
#define BEGIN (yy_start) = 1 + 2 *
```

10.12.1.3 ECHO

```
#define ECHO do { if (fwrite( yytext, (size_t) yylen, 1, yyout )) {} } while (0)
```

10.12.1.4 EOB_ACT_CONTINUE_SCAN

```
#define EOB_ACT_CONTINUE_SCAN 0
```

10.12.1.5 EOB_ACT_END_OF_FILE

```
#define EOB_ACT_END_OF_FILE 1
```

10.12.1.6 EOB_ACT_LAST_MATCH

```
#define EOB_ACT_LAST_MATCH 2
```

10.12.1.7 FLEX_BETA

```
#define FLEX_BETA
```

10.12.1.8 FLEX_DEBUG

```
#define FLEX_DEBUG
```

10.12.1.9 FLEX_SCANNER

```
#define FLEX_SCANNER
```

10.12.1.10 FLEXINT_H

```
#define FLEXINT_H
```

10.12.1.11 INITIAL

```
#define INITIAL 0
```

10.12.1.12 INT16_MAX

```
#define INT16_MAX (32767)
```

10.12.1.13 INT16_MIN

```
#define INT16_MIN (-32767-1)
```

10.12.1.14 INT32_MAX

```
#define INT32_MAX (2147483647)
```

10.12.1.15 INT32_MIN

```
#define INT32_MIN (-2147483647-1)
```

10.12.1.16 INT8_MAX

```
#define INT8_MAX (127)
```

10.12.1.17 INT8_MIN

```
#define INT8_MIN (-128)
```

10.12.1.18 REJECT

```
#define REJECT reject_used_but_not_detected
```

10.12.1.19 SIZE_MAX

```
#define SIZE_MAX (~ (size_t) 0)
```

10.12.1.20 UINT16_MAX

```
#define UINT16_MAX (65535U)
```

10.12.1.21 UINT32_MAX

```
#define UINT32_MAX (4294967295U)
```

10.12.1.22 UINT8_MAX

```
#define UINT8_MAX (255U)
```

10.12.1.23 unput

```
#define unput(  
    c ) yyunput( c, (yytext_ptr) )
```

10.12.1.24 YY_AT_BOL

```
#define YY_AT_BOL( ) (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)
```

10.12.1.25 YY_BREAK

```
#define YY_BREAK /*LINTED*/break;
```

10.12.1.26 YY_BUF_SIZE

```
#define YY_BUF_SIZE 16384
```

10.12.1.27 YY_BUFFER_EOF_PENDING

```
#define YY_BUFFER_EOF_PENDING 2
```

10.12.1.28 YY_BUFFER_NEW

```
#define YY_BUFFER_NEW 0
```

10.12.1.29 YY_BUFFER_NORMAL

```
#define YY_BUFFER_NORMAL 1
```

10.12.1.30 YY_CURRENT_BUFFER

```
#define YY_CURRENT_BUFFER
```

Value:

```
( (yy_buffer_stack) \
? (yy_buffer_stack)[(yy_buffer_stack_top)] \
: NULL)
```

10.12.1.31 YY_CURRENT_BUFFER_LVALUE

```
#define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]
```

10.12.1.32 YY_DO_BEFORE_ACTION

```
#define YY_DO_BEFORE_ACTION
```

Value:

```
(yytext_ptr) = yy_bp; \
/* %% [2.0] code to fiddle yytext and yyleng for yymore() goes here \ */\
(yytext_ptr) -= (yy_more_len); \
yyleng = (int) (yy_cp - (yytext_ptr)); \
(yy_hold_char) = *yy_cp; \
*yy_cp = '\0'; \
/* %% [3.0] code to copy yytext_ptr to yytext[] goes here, if %array \ */\
(yy_c_buf_p) = yy_cp;
```


10.12.1.33 YY_END_OF_BUFFER

```
#define YY_END_OF_BUFFER 28
```

10.12.1.34 YY_END_OF_BUFFER_CHAR

```
#define YY_END_OF_BUFFER_CHAR 0
```

10.12.1.35 YY_EXIT_FAILURE

```
#define YY_EXIT_FAILURE 2
```

10.12.1.36 YY_EXTRA_TYPE

```
#define YY_EXTRA_TYPE void *
```

10.12.1.37 YY_FATAL_ERROR

```
#define YY_FATAL_ERROR(  
    msg ) yy_fatal_error( msg )
```

10.12.1.38 YY_FLEX_MAJOR_VERSION

```
#define YY_FLEX_MAJOR_VERSION 2
```

10.12.1.39 YY_FLEX_MINOR_VERSION

```
#define YY_FLEX_MINOR_VERSION 6
```

10.12.1.40 YY_FLEX_SUBMINOR_VERSION

```
#define YY_FLEX_SUBMINOR_VERSION 4
```

10.12.1.41 YY_FLUSH_BUFFER

```
#define YY_FLUSH_BUFFER yy_flush_buffer( YY_CURRENT_BUFFER )
```

10.12.1.42 YY_INPUT

```
#define YY_INPUT(  
    buf,  
    result,  
    max_size )
```

10.12.1.43 YY_INT_ALIGNED

```
#define YY_INT_ALIGNED short int
```

10.12.1.44 YY_LESS_LINENO

```
#define YY_LESS_LINENO(  
    n )
```

10.12.1.45 YY_LINENO_REWIND_TO

```
#define YY_LINENO_REWIND_TO(  
    ptr )
```

10.12.1.46 YY_MORE_ADJ

```
#define YY_MORE_ADJ (yy_more_len)
```

10.12.1.47 yy_new_buffer

```
#define yy_new_buffer yy_create_buffer
```

10.12.1.48 YY_NEW_FILE

```
#define YY_NEW_FILE yyrestart( yyin )
```

10.12.1.49 YY_NO_INPUT

```
#define YY_NO_INPUT 1
```

10.12.1.50 YY_NULL

```
#define YY_NULL 0
```

10.12.1.51 YY_NUM_RULES

```
#define YY_NUM_RULES 27
```

10.12.1.52 YY_READ_BUF_SIZE

```
#define YY_READ_BUF_SIZE 8192
```

10.12.1.53 YY_RESTORE_YY_MORE_OFFSET

```
#define YY_RESTORE_YY_MORE_OFFSET
```

10.12.1.54 YY_RULE_SETUP

```
#define YY_RULE_SETUP YY_USER_ACTION
```

10.12.1.55 YY_SC_TO_UI

```
#define YY_SC_TO_UI(  
    c ) ((YY_CHAR) (c))
```

10.12.1.56 yy_set_bol

```
#define yy_set_bol(  
    at_bol )
```

Value:

```
{ \n  if ( ! YY_CURRENT_BUFFER ){ \n    yyensure_buffer_stack (); \n    YY_CURRENT_BUFFER_LVALUE = \n      yy_create_buffer( yyin, YY_BUF_SIZE ); \n  } \n  YY_CURRENT_BUFFER_LVALUE->yy_at_bol = at_bol; \n}
```

10.12.1.57 yy_set_interactive

```
#define yy_set_interactive(  
    is_interactive )
```

Value:

```
{ \n  if ( ! YY_CURRENT_BUFFER ){ \n    yyensure_buffer_stack (); \n    YY_CURRENT_BUFFER_LVALUE = \n      yy_create_buffer( yyin, YY_BUF_SIZE ); \n  } \n  YY_CURRENT_BUFFER_LVALUE->yy_is_interactive = is_interactive; \n}
```

10.12.1.58 YY_SKIP_YYWRAP

```
#define YY_SKIP_YYWRAP
```

10.12.1.59 YY_START

```
#define YY_START ((yy_start) - 1) / 2)
```

10.12.1.60 YY_START_STACK_INCR

```
#define YY_START_STACK_INCR 25
```

10.12.1.61 YY_STATE_BUF_SIZE

```
#define YY_STATE_BUF_SIZE ((YY_BUF_SIZE + 2) * sizeof(yy_state_type))
```

10.12.1.62 YY_STATE_EOF

```
#define YY_STATE_EOF(  
    state ) (YY_END_OF_BUFFER + state + 1)
```

10.12.1.63 YY_STRUCT_Y_BUFFER_STATE

```
#define YY_STRUCT_Y_BUFFER_STATE
```

10.12.1.64 YY_TYPEDEF_Y_BUFFER_STATE

```
#define YY_TYPEDEF_Y_BUFFER_STATE
```

10.12.1.65 YY_TYPEDEF_Y_SIZE_T

```
#define YY_TYPEDEF_Y_SIZE_T
```

10.12.1.66 YY_USER_ACTION

```
#define YY_USER_ACTION
```

Value:

```
loc.begin.line = loc.end.line; \  
loc.begin.column = loc.end.column; \  
for(int i = 0; yytext[i] != '\0'; i++) { \  
    if(yytext[i] == '\n') { \  
        loc.end.line++; \  
        loc.end.column = 0; \  
    } \  
    else { \  
        loc.end.column++; \  
    } \  
}
```

10.12.1.67 yyconst

```
#define yyconst const
```

10.12.1.68 yyless [1/2]

```
#define yyless(  
    n )
```

Value:

```
do \
{ \
    /* Undo effects of setting up yytext. */ \
    int yyless_macro_arg = (n); \
    YY_LESS_LINENO(yyless_macro_arg); \
    *yy_cp = (yy_hold_char); \
    YY_RESTORE_YY_MORE_OFFSET \
    (yy_c_buf_p) = yy_cp = yy_bp + yyless_macro_arg - YY_MORE_ADJ; \
    YY_DO_BEFORE_ACTION; /* set up yytext again */ \
} \
while ( 0 )
```

10.12.1.69 yyless [2/2]

```
#define yyless(  
    n )
```

Value:

```
do \
{ \
    /* Undo effects of setting up yytext. */ \
    int yyless_macro_arg = (n); \
    YY_LESS_LINENO(yyless_macro_arg); \
    yytext[yy_leng] = (yy_hold_char); \
    (yy_c_buf_p) = yytext + yyless_macro_arg; \
    (yy_hold_char) = *(yy_c_buf_p); \
    *(yy_c_buf_p) = '\0'; \
    yy_leng = yyless_macro_arg; \
} \
while ( 0 )
```

10.12.1.70 yymore

```
#define yymore( ) ((yy_more_flag) = 1)
```

10.12.1.71 yynoreturn

```
#define yynoreturn
```

10.12.1.72 YYSTATE

```
#define YYSTATE YY_START
```

10.12.1.73 YYTABLES_NAME

```
#define YYTABLES_NAME "yytables"
```

10.12.1.74 yyterminate

```
#define yyterminate( ) return YY_NULL
```

10.12.1.75 yytext_ptr

```
#define yytext_ptr yytext
```

10.12.1.76 yywrap

```
#define yywrap( ) (/*CONSTCOND*/1)
```

10.12.2 Typedef Documentation

10.12.2.1 flex_int16_t

```
typedef short int flex_int16_t
```

10.12.2.2 flex_int32_t

```
typedef int flex_int32_t
```

10.12.2.3 flex_int8_t

```
typedef signed char flex_int8_t
```

10.12.2.4 flex_uint16_t

```
typedef unsigned short int flex_uint16_t
```

10.12.2.5 flex_uint32_t

```
typedef unsigned int flex_uint32_t
```

10.12.2.6 flex_uint8_t

```
typedef unsigned char flex_uint8_t
```

10.12.2.7 YY_BUFFER_STATE

```
typedef struct yy_buffer_state* YY_BUFFER_STATE
```

10.12.2.8 YY_CHAR

```
typedef flex_uint8_t YY_CHAR
```

10.12.2.9 yy_size_t

```
typedef size_t yy_size_t
```

10.12.2.10 yy_state_type

```
typedef int yy_state_type
```


10.12.3 Function Documentation

10.12.3.1 if()

```
if (
    ! yy_init )
```

10.12.3.2 step()

```
loc step ( )
```

10.12.3.3 while()

```
while (
    1 )
```

10.12.3.4 yy_create_buffer()

```
YY_BUFFER_STATE yy_create_buffer (
    FILE * file,
    int size )
```

Allocate and initialize an input buffer state.

Parameters

<i>file</i>	A readable stream.
<i>size</i>	The character buffer size in bytes. When in doubt, use YY_BUF_SIZE.

Returns

the allocated buffer state.

10.12.3.5 yy_delete_buffer()

```
void yy_delete_buffer (
    YY_BUFFER_STATE b )
```

Destroy the buffer.

Parameters

<i>b</i>	a buffer created with yy_create_buffer()
----------	--

10.12.3.6 yy_flush_buffer()

```
void yy_flush_buffer (
    YY_BUFFER_STATE b )
```

Discard all buffered characters. On the next scan, YY_INPUT will be called.

Parameters

<i>b</i>	the buffer state to be flushed, usually YY_CURRENT_BUFFER.
----------	--

10.12.3.7 yy_scan_buffer()

```
YY_BUFFER_STATE yy_scan_buffer (
    char * base,
    yy_size_t size )
```

Setup the input buffer state to scan directly from a user-specified character buffer.

Parameters

<i>base</i>	the character buffer
<i>size</i>	the size in bytes of the character buffer

Returns

the newly allocated buffer state object.

10.12.3.8 yy_scan_bytes()

```
YY_BUFFER_STATE yy_scan_bytes (
    const char * yybytes,
    int _yybytes_len )
```

Setup the input buffer state to scan the given bytes. The next call to [yylex\(\)](#) will scan from a *copy* of *bytes*.

Parameters

<code>yybytes</code>	the byte buffer to scan
<code>_yybytes_len</code>	the number of bytes in the buffer pointed to by <i>bytes</i> .

Returns

the newly allocated buffer state object.

10.12.3.9 yy_scan_string()

```
YY_BUFFER_STATE yy_scan_string (  
    const char * yystr )
```

Setup the input buffer state to scan a string. The next call to [yylex\(\)](#) will scan from a *copy* of *str*.

Parameters

<code>yystr</code>	a NUL-terminated string to scan
--------------------	---------------------------------

Returns

the newly allocated buffer state object.

Note

If you want to scan bytes that may contain NUL values, then use [yy_scan_bytes\(\)](#) instead.

10.12.3.10 yy_switch_to_buffer()

```
void yy_switch_to_buffer (  
    YY_BUFFER_STATE new_buffer )
```

Switch to a different input buffer.

Parameters

<code>new_buffer</code>	The new input buffer.
-------------------------	-----------------------

10.12.3.11 yyalloc()

```
void* yyalloc (
    yy_size_t size )
```

10.12.3.12 yyfree()

```
void yyfree (
    void * ptr )
```

10.12.3.13 yyget_debug()

```
int yyget_debug (
    void )
```

10.12.3.14 yyget_extra()

```
YY_EXTRA_TYPE yyget_extra (
    void )
```

10.12.3.15 yyget_in()

```
FILE* yyget_in (
    void )
```

Get the input stream.

10.12.3.16 yyget_leng()

```
int yyget_leng (
    void )
```

Get the length of the current token.

10.12.3.17 yyget_lineno()

```
int yyget_lineno (
    void )
```

Get the current line number.

10.12.3.18 yyget_out()

```
FILE* yyget_out (
    void )
```

Get the output stream.

10.12.3.19 yyget_text()

```
char* yyget_text (
    void )
```

Get the current token.

10.12.3.20 yylex_destroy()

```
int yylex_destroy (
    void )
```

10.12.3.21 yypop_buffer_state()

```
void yypop_buffer_state (
    void )
```

Removes and deletes the top of the stack, if present. The next element becomes the new top.

10.12.3.22 yypush_buffer_state()

```
void yypush_buffer_state (
    YY_BUFFER_STATE new_buffer )
```

Pushes the new state onto the stack. The new state becomes the current state. This function will allocate the stack if necessary.

Parameters

<i>new_buffer</i>	The new state.
-------------------	----------------

10.12.3.23 yyrealloc()

```
void* yyrealloc (
    void * ptr,
    yy_size_t size )
```

10.12.3.24 yyrestart()

```
void yyrestart (
    FILE * input_file )
```

Immediately switch to a different input stream.

Parameters

<i>input_file</i>	A readable stream.
-------------------	--------------------

Note

This function does not reset the start condition to `INITIAL`.

10.12.3.25 yyset_debug()

```
void yyset_debug (
    int debug_flag )
```

10.12.3.26 yyset_extra()

```
void yyset_extra (
    YY_EXTRA_TYPE user_defined )
```

10.12.3.27 yyset_in()

```
void yyset_in (
    FILE * _in_str )
```

Set the input stream. This does not discard the current input buffer.

Parameters

<i>_in_str</i>	A readable stream.
----------------	--------------------

See also

[yy_switch_to_buffer](#)

10.12.3.28 yyset_lineno()

```
void yyset_lineno (
    int _line_number )
```

Set the current line number.

Parameters

<code>_line_number</code>	line number
---------------------------	-------------

10.12.3.29 yyset_out()

```
void yyset_out (
    FILE * _out_str )
```

10.12.4 Variable Documentation

10.12.4.1 subloc

```
yy::location subloc
```

Initial value:

```
{
    yy::location& loc = drv.get_location()
```

10.12.4.2 yy_act

```
int yy_act
```

10.12.4.3 yy_bp

```
char * yy_bp
```

10.12.4.4 yy_cp

```
char* yy_cp
```

10.12.4.5 YY_DECL

```
YY_DECL
```

Initial value:

```
{  
    yy_state_type yy_current_state
```

The main scanner function which does all the work.

10.12.4.6 yy_flex_debug

```
int yy_flex_debug = 1
```

10.12.4.7 yyin

```
FILE* yyin = NULL
```

10.12.4.8 yyleng

```
int yyleng
```

10.12.4.9 yylineno

```
int yylineno = 1
```

10.12.4.10 yyout

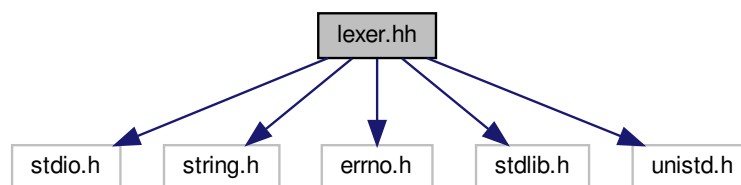
```
FILE * yyout = NULL
```


10.12.4.11 yytext

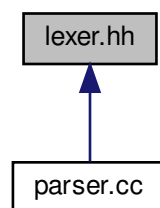
```
char* yytext
```

10.13 lexer.hh File Reference

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
Include dependency graph for lexer.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [yy_buffer_state](#)

Macros

- `#define yyIN_HEADER 1`
- `#define YY_INT_ALIGNED short int`
- `#define FLEX_SCANNER`
- `#define YY_FLEX_MAJOR_VERSION 2`
- `#define YY_FLEX_MINOR_VERSION 6`
- `#define YY_FLEX_SUBMINOR_VERSION 4`
- `#define FLEX_BETA`
- `#define FLEXINT_H`
- `#define INT8_MIN (-128)`
- `#define INT16_MIN (-32767-1)`
- `#define INT32_MIN (-2147483647-1)`
- `#define INT8_MAX (127)`
- `#define INT16_MAX (32767)`
- `#define INT32_MAX (2147483647)`
- `#define UINT8_MAX (255U)`
- `#define UINT16_MAX (65535U)`
- `#define UINT32_MAX (4294967295U)`
- `#define SIZE_MAX (~(size_t)0)`
- `#define yyconst const`
- `#define yynoreturn`
- `#define YY_BUF_SIZE 16384`
- `#define YY_TYPEDEF_YY_BUFFER_STATE`
- `#define YY_TYPEDEF_YY_SIZE_T`
- `#define YY_STRUCT_YY_BUFFER_STATE`
- `#define yywrap() (/*CONSTCOND*/1)`
- `#define YY_SKIP_YYWRAP`
- `#define FLEX_DEBUG`
- `#define yytext_ptr yytext`
- `#define YY_EXTRA_TYPE void *`
- `#define YY_READ_BUF_SIZE 8192`
- `#define YY_START_STACK_INCR 25`
- `#define YY_DECL_IS_OURS 1`
- `#define YY_DECL int yylex (void)`

Typedefs

- `typedef signed char flex_int8_t`
- `typedef short int flex_int16_t`
- `typedef int flex_int32_t`
- `typedef unsigned char flex_uint8_t`
- `typedef unsigned short int flex_uint16_t`
- `typedef unsigned int flex_uint32_t`
- `typedef struct yy_buffer_state * YY_BUFFER_STATE`
- `typedef size_t yy_size_t`

Functions

- void [yyrestart](#) (FILE *input_file)
- void [yy_switch_to_buffer](#) (YY_BUFFER_STATE new_buffer)
- YY_BUFFER_STATE [yy_create_buffer](#) (FILE *file, int size)
- void [yy_delete_buffer](#) (YY_BUFFER_STATE b)
- void [yy_flush_buffer](#) (YY_BUFFER_STATE b)
- void [yypush_buffer_state](#) (YY_BUFFER_STATE new_buffer)
- void [yypop_buffer_state](#) (void)
- YY_BUFFER_STATE [yy_scan_buffer](#) (char *base, [yy_size_t](#) size)
- YY_BUFFER_STATE [yy_scan_string](#) (const char *yy_str)
- YY_BUFFER_STATE [yy_scan_bytes](#) (const char *bytes, int len)
- void * [yyalloc](#) ([yy_size_t](#))
- void * [yyrealloc](#) (void *, [yy_size_t](#))
- void [yyfree](#) (void *)
- int [yylex_destroy](#) (void)
- int [yyget_debug](#) (void)
- void [yyset_debug](#) (int debug_flag)
- YY_EXTRA_TYPE [yyget_extra](#) (void)
- void [yyset_extra](#) (YY_EXTRA_TYPE user_defined)
- FILE * [yyget_in](#) (void)
- void [yyset_in](#) (FILE *_in_str)
- FILE * [yyget_out](#) (void)
- void [yyset_out](#) (FILE *_out_str)
- int [yyget_leng](#) (void)
- char * [yyget_text](#) (void)
- int [yyget_lineno](#) (void)
- void [yyset_lineno](#) (int _line_number)
- int [yylex](#) (void)

Variables

- int [yyleng](#)
- FILE * [yyin](#)
- FILE * [yyout](#)
- int [yylineno](#)
- char * [yytext](#)

10.13.1 Macro Definition Documentation

10.13.1.1 FLEX_BETA

```
#define FLEX_BETA
```

10.13.1.2 FLEX_DEBUG

```
#define FLEX_DEBUG
```

10.13.1.3 FLEX_SCANNER

```
#define FLEX_SCANNER
```

10.13.1.4 FLEXINT_H

```
#define FLEXINT_H
```

10.13.1.5 INT16_MAX

```
#define INT16_MAX (32767)
```

10.13.1.6 INT16_MIN

```
#define INT16_MIN (-32767-1)
```

10.13.1.7 INT32_MAX

```
#define INT32_MAX (2147483647)
```

10.13.1.8 INT32_MIN

```
#define INT32_MIN (-2147483647-1)
```

10.13.1.9 INT8_MAX

```
#define INT8_MAX (127)
```

10.13.1.10 INT8_MIN

```
#define INT8_MIN (-128)
```

10.13.1.11 SIZE_MAX

```
#define SIZE_MAX (~(size_t)0)
```

10.13.1.12 UINT16_MAX

```
#define UINT16_MAX (65535U)
```

10.13.1.13 UINT32_MAX

```
#define UINT32_MAX (4294967295U)
```

10.13.1.14 UINT8_MAX

```
#define UINT8_MAX (255U)
```

10.13.1.15 YY_BUF_SIZE

```
#define YY_BUF_SIZE 16384
```

10.13.1.16 YY_DECL

```
#define YY_DECL int yylex (void)
```

10.13.1.17 YY_DECL_IS_OURS

```
#define YY_DECL_IS_OURS 1
```

10.13.1.18 YY_EXTRA_TYPE

```
#define YY_EXTRA_TYPE void *
```

10.13.1.19 YY_FLEX_MAJOR_VERSION

```
#define YY_FLEX_MAJOR_VERSION 2
```

10.13.1.20 YY_FLEX_MINOR_VERSION

```
#define YY_FLEX_MINOR_VERSION 6
```

10.13.1.21 YY_FLEX_SUBMINOR_VERSION

```
#define YY_FLEX_SUBMINOR_VERSION 4
```

10.13.1.22 YY_INT_ALIGNED

```
#define YY_INT_ALIGNED short int
```

10.13.1.23 YY_READ_BUF_SIZE

```
#define YY_READ_BUF_SIZE 8192
```

10.13.1.24 YY_SKIP_YYWRAP

```
#define YY_SKIP_YYWRAP
```

10.13.1.25 YY_START_STACK_INCR

```
#define YY_START_STACK_INCR 25
```

10.13.1.26 YY_STRUCT_YY_BUFFER_STATE

```
#define YY_STRUCT_YY_BUFFER_STATE
```

10.13.1.27 YY_TYPEDEF_YY_BUFFER_STATE

```
#define YY_TYPEDEF_YY_BUFFER_STATE
```

10.13.1.28 YY_TYPEDEF_YY_SIZE_T

```
#define YY_TYPEDEF_YY_SIZE_T
```

10.13.1.29 yyconst

```
#define yyconst const
```

10.13.1.30 yyIN_HEADER

```
#define yyIN_HEADER 1
```

10.13.1.31 yynoreturn

```
#define yynoreturn
```

10.13.1.32 yytext_ptr

```
#define yytext_ptr yytext
```

10.13.1.33 yywrap

```
#define yywrap( ) (/*CONSTCOND*/1)
```

10.13.2 Typedef Documentation

10.13.2.1 flex_int16_t

```
typedef short int flex_int16_t
```

10.13.2.2 flex_int32_t

```
typedef int flex_int32_t
```

10.13.2.3 flex_int8_t

```
typedef signed char flex_int8_t
```

10.13.2.4 flex_uint16_t

```
typedef unsigned short int flex_uint16_t
```

10.13.2.5 flex_uint32_t

```
typedef unsigned int flex_uint32_t
```

10.13.2.6 flex_uint8_t

```
typedef unsigned char flex_uint8_t
```

10.13.2.7 YY_BUFFER_STATE

```
typedef struct yy_buffer_state* YY_BUFFER_STATE
```


10.13.2.8 yy_size_t

```
typedef size_t yy_size_t
```

10.13.3 Function Documentation

10.13.3.1 yy_create_buffer()

```
YY_BUFFER_STATE yy_create_buffer (
    FILE * file,
    int size )
```

Allocate and initialize an input buffer state.

Parameters

<i>file</i>	A readable stream.
<i>size</i>	The character buffer size in bytes. When in doubt, use YY_BUF_SIZE.

Returns

the allocated buffer state.

10.13.3.2 yy_delete_buffer()

```
void yy_delete_buffer (
    YY_BUFFER_STATE b )
```

Destroy the buffer.

Parameters

<i>b</i>	a buffer created with yy_create_buffer()
----------	--

10.13.3.3 yy_flush_buffer()

```
void yy_flush_buffer (
    YY_BUFFER_STATE b )
```

Discard all buffered characters. On the next scan, YY_INPUT will be called.

Parameters

<i>b</i>	the buffer state to be flushed, usually <code>YY_CURRENT_BUFFER</code> .
----------	--

10.13.3.4 yy_scan_buffer()

```
YY_BUFFER_STATE yy_scan_buffer (
    char * base,
    yy_size_t size )
```

Setup the input buffer state to scan directly from a user-specified character buffer.

Parameters

<i>base</i>	the character buffer
<i>size</i>	the size in bytes of the character buffer

Returns

the newly allocated buffer state object.

10.13.3.5 yy_scan_bytes()

```
YY_BUFFER_STATE yy_scan_bytes (
    const char * bytes,
    int len )
```

Setup the input buffer state to scan the given bytes. The next call to `yylex()` will scan from a *copy* of *bytes*.

Parameters

<i>yybytes</i>	the byte buffer to scan
<i>_yybytes_len</i>	the number of bytes in the buffer pointed to by <i>bytes</i> .

Returns

the newly allocated buffer state object.

10.13.3.6 yy_scan_string()

```
YY_BUFFER_STATE yy_scan_string (
    const char * yy_str )
```

Setup the input buffer state to scan a string. The next call to `yylex()` will scan from a *copy* of *str*.

Parameters

<i>yystr</i>	a NUL-terminated string to scan
--------------	---------------------------------

Returns

the newly allocated buffer state object.

Note

If you want to scan bytes that may contain NUL values, then use [yy_scan_bytes\(\)](#) instead.

10.13.3.7 yy_switch_to_buffer()

```
void yy_switch_to_buffer (
    YY_BUFFER_STATE new_buffer )
```

Switch to a different input buffer.

Parameters

<i>new_buffer</i>	The new input buffer.
-------------------	-----------------------

10.13.3.8 yyallocc()

```
void* yyallocc (
    yy_size_t size )
```

10.13.3.9 yyfree()

```
void yyfree (
    void * ptr )
```

10.13.3.10 yyget_debug()

```
int yyget_debug (
    void )
```

10.13.3.11 yyget_extra()

```
YY_EXTRA_TYPE yyget_extra (  
    void )
```

10.13.3.12 yyget_in()

```
FILE* yyget_in (  
    void )
```

Get the input stream.

10.13.3.13 yyget_leng()

```
int yyget_leng (  
    void )
```

Get the length of the current token.

10.13.3.14 yyget_lineno()

```
int yyget_lineno (  
    void )
```

Get the current line number.

10.13.3.15 yyget_out()

```
FILE* yyget_out (  
    void )
```

Get the output stream.

10.13.3.16 yyget_text()

```
char* yyget_text (  
    void )
```

Get the current token.

10.13.3.17 yylex()

```
int yylex (  
    void )
```

10.13.3.18 yylex_destroy()

```
int yylex_destroy (
    void )
```

10.13.3.19 yypop_buffer_state()

```
void yypop_buffer_state (
    void )
```

Removes and deletes the top of the stack, if present. The next element becomes the new top.

10.13.3.20 yypush_buffer_state()

```
void yypush_buffer_state (
    YY_BUFFER_STATE new_buffer )
```

Pushes the new state onto the stack. The new state becomes the current state. This function will allocate the stack if necessary.

Parameters

<i>new_buffer</i>	The new state.
-------------------	----------------

10.13.3.21 yyrealloc()

```
void* yyrealloc (
    void * ptr,
    yy_size_t size )
```

10.13.3.22 yyrestart()

```
void yyrestart (
    FILE * input_file )
```

Immediately switch to a different input stream.

Parameters

<i>input_file</i>	A readable stream.
-------------------	--------------------

Note

This function does not reset the start condition to `INITIAL`.

10.13.3.23 yyset_debug()

```
void yyset_debug (
    int debug_flag )
```

10.13.3.24 yyset_extra()

```
void yyset_extra (
    YY\_EXTRA\_TYPE user_defined )
```

10.13.3.25 yyset_in()

```
void yyset_in (
    FILE * _in_str )
```

Set the input stream. This does not discard the current input buffer.

Parameters

<code>_in_str</code>	A readable stream.
----------------------	--------------------

See also

[yy_switch_to_buffer](#)

10.13.3.26 yyset_lineno()

```
void yyset_lineno (
    int _line_number )
```

Set the current line number.

Parameters

<code>_line_number</code>	line number
---------------------------	-------------

10.13.3.27 yyset_out()

```
void yyset_out (
    FILE * _out_str )
```

10.13.4 Variable Documentation

10.13.4.1 yyin

```
FILE* yyin [extern]
```

10.13.4.2 yyleng

```
int yyleng [extern]
```

10.13.4.3 yylineno

```
int yylineno [extern]
```

10.13.4.4 yyout

```
FILE * yyout
```

10.13.4.5 yytext

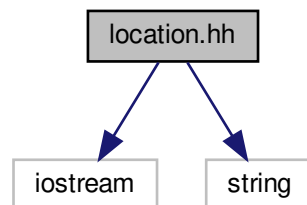
```
char* yytext [extern]
```

10.14 location.hh File Reference

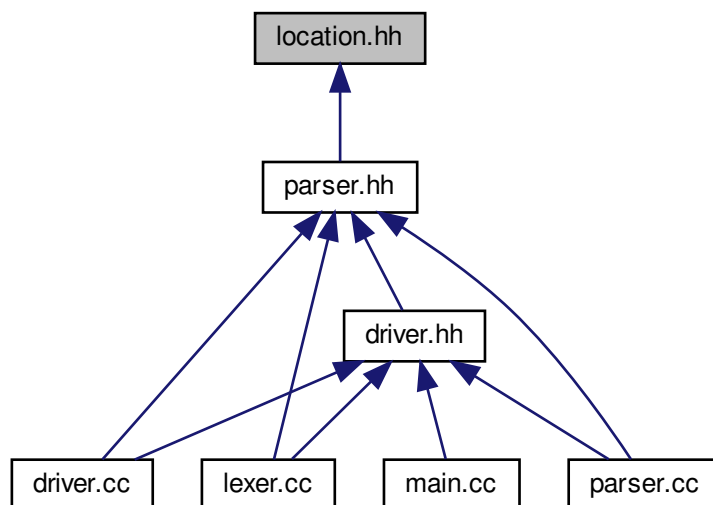
```
#include <iostream>
```

```
#include <string>
```

Include dependency graph for location.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class [yy::position](#)
A point in a source file.
- class [yy::location](#)
Two points in a source file.

Namespaces

- [yy](#)

Macros

- `#define YY_NULLPTR ((void*)0)`

Functions

- `position & yy::operator+= (position &res, position::counter_type width)`
Add width columns, in place.
- `position yy::operator+ (position res, position::counter_type width)`
Add width columns.
- `position & yy::operator-= (position &res, position::counter_type width)`
Subtract width columns, in place.
- `position yy::operator- (position res, position::counter_type width)`
Subtract width columns.
- `template<typename YYChar > std::basic_ostream< YYChar > & yy::operator<< (std::basic_ostream< YYChar > &ostr, const position &pos)`
Intercept output stream redirection.
- `location & yy::operator+= (location &res, const location &end)`
Join two locations, in place.
- `location yy::operator+ (location res, const location &end)`
Join two locations.
- `location & yy::operator+= (location &res, location::counter_type width)`
Add width columns to the end position, in place.
- `location yy::operator+ (location res, location::counter_type width)`
Add width columns to the end position.
- `location & yy::operator-= (location &res, location::counter_type width)`
Subtract width columns to the end position, in place.
- `location yy::operator- (location res, location::counter_type width)`
Subtract width columns to the end position.
- `template<typename YYChar > std::basic_ostream< YYChar > & yy::operator<< (std::basic_ostream< YYChar > &ostr, const location &loc)`
Intercept output stream redirection.

10.14.1 Detailed Description

Define the [yy::location](#) class.

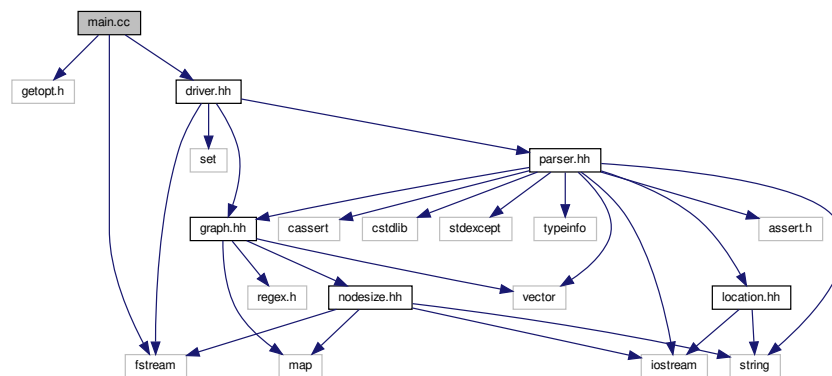
10.14.2 Macro Definition Documentation

10.14.2.1 YY_NULLPTR

```
#define YY_NULLPTR ((void*)0)
```

10.15 main.cc File Reference

```
#include <getopt.h>
#include <fstream>
#include <driver.hh>
Include dependency graph for main.cc:
```



Functions

- void [usage](#) (char *name)
- int [main](#) (int argc, char **argv)

Variables

- struct option [options](#) []
- char * [optstring](#) = (char*)"nfh"
- char * [description](#) []

10.15.1 Function Documentation

10.15.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

10.15.1.2 usage()

```
void usage (
    char * name )
```

10.15.2 Variable Documentation

10.15.2.1 description

```
char* description[]
```

Initial value:

```
= {
    (char*)"    Do not do any graph transformations.",
    (char*)"    Wrap all tikzpictures in figures and create commands"
    "to place them.",
    (char*)"",
    (char*)" "
}
```

10.15.2.2 options

```
struct option options[]
```

Initial value:

```
= {
    {"nooptimize", no_argument,  NULL, 'n'},
    {"makefigures", no_argument,  NULL, 'f'},
    {"help", no_argument,  NULL, 'h'},
    {0, 0, 0, 0}
}
```

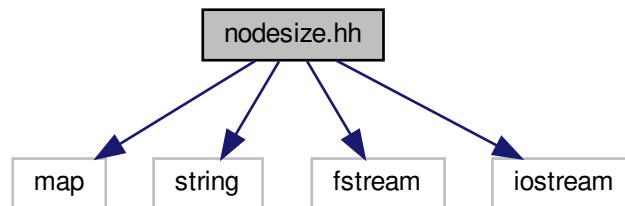
10.15.2.3 optstring

```
char* optstring = (char*)"nfh"
```

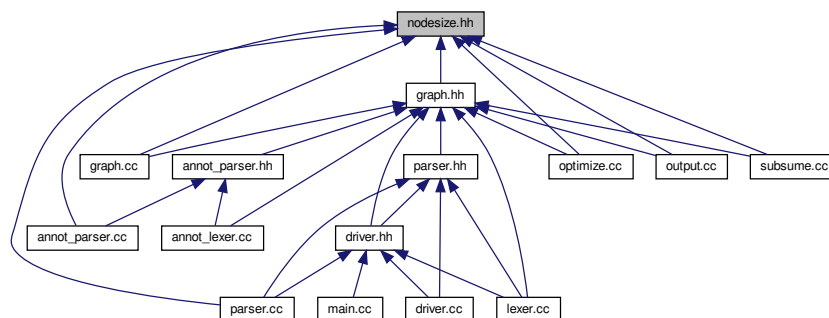
10.16 nodesize.hh File Reference

```
#include <map>
#include <string>
#include <fstream>
#include <iostream>
```

Include dependency graph for nodesize.hh:



This graph shows which files directly or indirectly include this file:



Classes

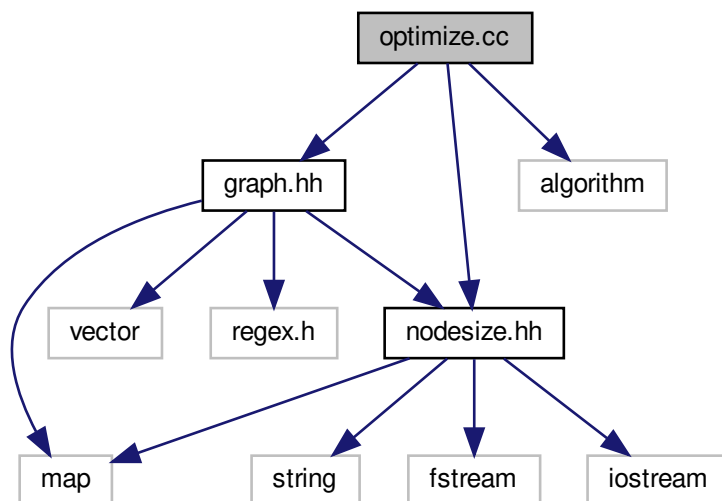
- class [coordinate](#)
- class [nodesizes](#)

10.17 optimize.cc File Reference

```
#include <graph.hh>
#include <nodesize.hh>
```

```
#include <algorithm>
```

Include dependency graph for optimize.cc:



10.18 output.cc File Reference

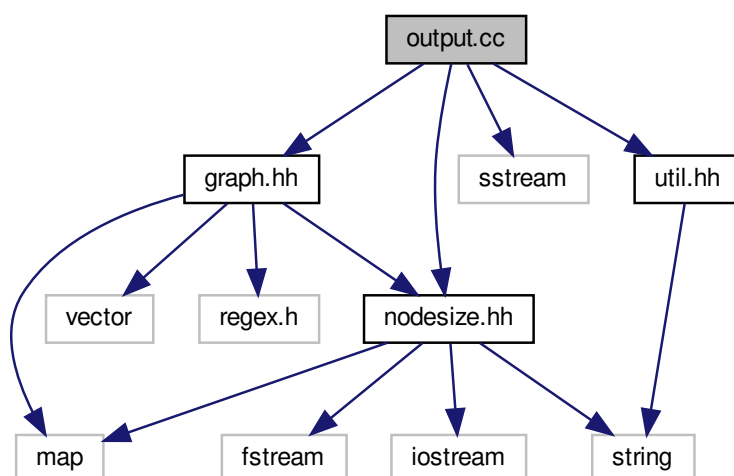
```
#include <graph.hh>
```

```
#include <sstream>
```

```
#include <nodesize.hh>
```

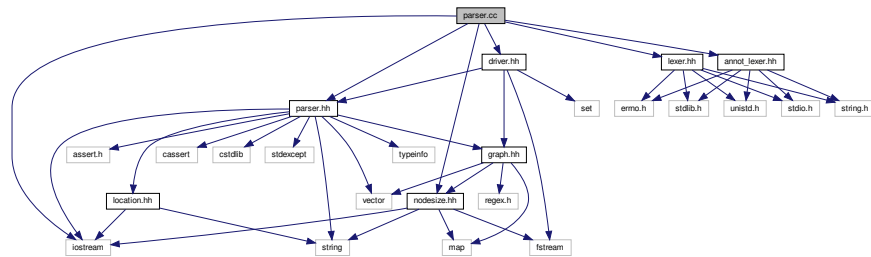
```
#include <util.hh>
```

Include dependency graph for output.cc:



10.19 parser.cc File Reference

```
#include "parser.hh"
#include <iostream>
#include "driver.hh"
#include "annot_lexer.hh"
#include "lexer.hh"
#include "nodesize.hh"
Include dependency graph for parser.cc:
```



Namespaces

- [yy](#)

Macros

- `#define YY_(msgid) msgid`
- `#define YY_EXCEPTIONS 1`
- `#define YYRHSLOC(Rhs, K) ((Rhs)[K].location)`
- `#define YYLLOC_DEFAULT(Current, Rhs, N)`
- `#define YYCDEBUG if (yydebug_) (*yycdebug_)`
- `#define YY_SYMBOL_PRINT(Title, Symbol)`
- `#define YY_REDUCE_PRINT(Rule)`
- `#define YY_STACK_PRINT()`
- `#define yyerrok (yyerrstatus_ = 0)`
- `#define yyclearin (yyla.clear ())`
- `#define YYACCEPT goto yyacceptlab`
- `#define YYABORT goto yyabortlab`
- `#define YYERROR goto yyerrorlab`
- `#define YYRECOVERING() (!yyerrstatus_)`
- `#define YYCASE_(N, S)`

Functions

- `node * wrapChoice (node *n)`

Variables

- `yy::location loc`

10.19.1 Macro Definition Documentation

10.19.1.1 YY_

```
#define YY_(  
    msgid ) msgid
```

10.19.1.2 YY_EXCEPTIONS

```
#define YY_EXCEPTIONS 1
```

10.19.1.3 YY_REDUCE_PRINT

```
#define YY_REDUCE_PRINT(  
    Rule )
```

Value:

```
do {  
    if (yydebug_)  
        yy_reduce_print_ (Rule);  
} while (false)
```

10.19.1.4 YY_STACK_PRINT

```
#define YY_STACK_PRINT( )
```

Value:

```
do {  
    if (yydebug_)  
        yy_stack_print_ ();  
} while (false)
```

10.19.1.5 YY_SYMBOL_PRINT

```
#define YY_SYMBOL_PRINT(  
    Title,  
    Symbol )
```

Value:

```
do {  
    if (yydebug_)  
    {  
        *yydebug_ « Title « ' ' ;  
        yy_print_ (*yydebug_, Symbol);  
        *yydebug_ « '\n';  
    }  
} while (false)
```

```
#define YYABORT goto yyabortlab
```

```
#define YYABORT goto yyabortlab
```

```
#define YYACCEPT goto yyacceptlab
```

```
#define YYACCEPT goto yyacceptlab
```

```
#define YYCASE_(  
    N,  
    S )
```

```
#define YYCASE_(  
    N,  
    S )
```

```
case N:
    yyformat = S;
break
```

/ /

```
#define YYCDEBUG if (yydebug_) (*yycdebug_)
```

```
#define YYCDEBUG if (yydebug_) (*yycdebug_)
```

```
#define yyclearin (yyla.clear ())
```

```
#define yyclearin (yyla.clear ())
```

```
#define yyerrok (yyerrstatus_ = 0)
```

```
#define yyerrok (yyerrstatus_ = 0)
```

```
#define YYERROR goto yyerrorlab
```

```
#define YYERROR goto yyerrorlab
```


10.19.1.13 YYLLOC_DEFAULT

```
#define YYLLOC_DEFAULT(  
    Current,  
    Rhs,  
    N )
```

Value:

```
do  
    if (N)  
    {  
        (Current).begin = YYRHSLOC (Rhs, 1).begin;  
        (Current).end   = YYRHSLOC (Rhs, N).end;  
    }  
    else  
    {  
        (Current).begin = (Current).end = YYRHSLOC (Rhs, 0).end;  
    }  
while (false)
```

10.19.1.14 YYRECOVERING

```
#define YYRECOVERING( ) (!yyerrstatus_)
```

10.19.1.15 YYRHSLOC

```
#define YYRHSLOC(  
    Rhs,  
    K ) ((Rhs)[K].location)
```

10.19.2 Function Documentation

10.19.2.1 wrapChoice()

```
node* wrapChoice (  
    node * n )
```

10.19.3 Variable Documentation

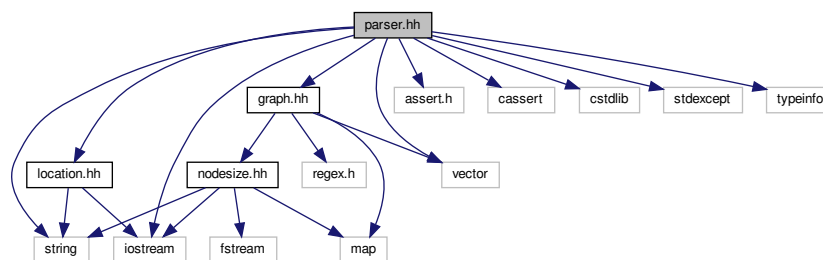
10.19.3.1 loc

```
yy::location loc [extern]
```

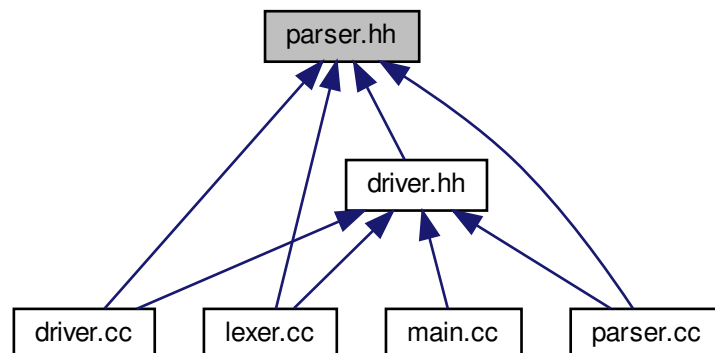
10.20 parser.hh File Reference

```
#include <string>
#include <assert.h>
#include "graph.hh"
#include <cassert>
#include <cstdlib>
#include <iostream>
#include <stdexcept>
#include <vector>
#include "location.hh"
#include <typeinfo>
```

Include dependency graph for parser.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class [yy::parser](#)
A Bison parser.
- class [yy::parser::value_type](#)
- struct [yy::parser::syntax_error](#)

Syntax errors thrown from user actions.

- struct [yy::parser::token](#)

Token kinds.

- struct [yy::parser::symbol_kind](#)

Symbol kinds.

- struct [yy::parser::basic_symbol< Base >](#)
- struct [yy::parser::by_kind](#)

Type access provider for token (enum) based symbols.

- struct [yy::parser::symbol_type](#)

"External" symbols: returned by the scanner.

- class [yy::parser::context](#)
- class [yy::parser::stack< T, S >::slice](#)

Present a slice of the top of a stack.

Namespaces

- [yy](#)

Macros

- `#define YY_CPLUSPLUS 199711L`
- `#define YY_MOVE`
- `#define YY_MOVE_OR_COPY copy`
- `#define YY_MOVE_REF(Type) Type&`
- `#define YY_RVREF(Type) const Type&`
- `#define YY_COPY(Type) const Type&`
- `#define YY_NOEXCEPT`
- `#define YY_NOTHROW throw ()`
- `#define YY_CONSTEXPR`
- `#define YY_ASSERT assert`
- `#define YY_ATTRIBUTE_PURE`
- `#define YY_ATTRIBUTE_UNUSED`
- `#define YY_USE(E) ((void) (E))`
- `#define YY_INITIAL_VALUE(Value) Value`
- `#define YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN`
- `#define YY_IGNORE_MAYBE_UNINITIALIZED_END`
- `#define YY_IGNORE_USELESS_CAST_BEGIN`
- `#define YY_IGNORE_USELESS_CAST_END`
- `#define YY_CAST(Type, Val) ((Type) (Val))`
- `#define YY_REINTERPRET_CAST(Type, Val) ((Type) (Val))`
- `#define YYDEBUG 1`

Functions

- [annotmap](#) * [scanAnnot](#) (string &s, void *loc)

10.20.1 Detailed Description

Define the [yy::parser](#) class.

10.20.2 Macro Definition Documentation

10.20.2.1 YY_ASSERT

```
#define YY_ASSERT assert
```

10.20.2.2 YY_ATTRIBUTE_PURE

```
#define YY_ATTRIBUTE_PURE
```

10.20.2.3 YY_ATTRIBUTE_UNUSED

```
#define YY_ATTRIBUTE_UNUSED
```

10.20.2.4 YY_CAST

```
#define YY_CAST(  
    Type,  
    Val ) ((Type) (Val))
```

10.20.2.5 YY_CONSTEXPR

```
#define YY_CONSTEXPR
```

10.20.2.6 YY_COPY

```
#define YY_COPY(  
    Type ) const Type&
```

10.20.2.7 YY_CPLUSPLUS

```
#define YY_CPLUSPLUS 199711L
```

10.20.2.8 YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN

```
#define YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN
```

10.20.2.9 YY_IGNORE_MAYBE_UNINITIALIZED_END

```
#define YY_IGNORE_MAYBE_UNINITIALIZED_END
```

10.20.2.10 YY_IGNORE_USELESS_CAST_BEGIN

```
#define YY_IGNORE_USELESS_CAST_BEGIN
```

10.20.2.11 YY_IGNORE_USELESS_CAST_END

```
#define YY_IGNORE_USELESS_CAST_END
```

10.20.2.12 YY_INITIAL_VALUE

```
#define YY_INITIAL_VALUE(  
    Value ) Value
```

10.20.2.13 YY_MOVE

```
#define YY_MOVE
```

10.20.2.14 YY_MOVE_OR_COPY

```
#define YY_MOVE_OR_COPY copy
```

10.20.2.15 YY_MOVE_REF

```
#define YY_MOVE_REF(  
    Type ) Type&
```

10.20.2.16 YY_NOEXCEPT

```
#define YY_NOEXCEPT
```

10.20.2.17 YY_NOTHROW

```
#define YY_NOTHROW throw ()
```

10.20.2.18 YY_REINTERPRET_CAST

```
#define YY_REINTERPRET_CAST(  
    Type,  
    Val ) ((Type) (Val))
```

10.20.2.19 YY_RVREF

```
#define YY_RVREF(  
    Type ) const Type&
```

10.20.2.20 YY_USE

```
#define YY_USE(  
    E ) ((void) (E))
```

10.20.2.21 YYDEBUG

```
#define YYDEBUG 1
```

10.20.3 Function Documentation

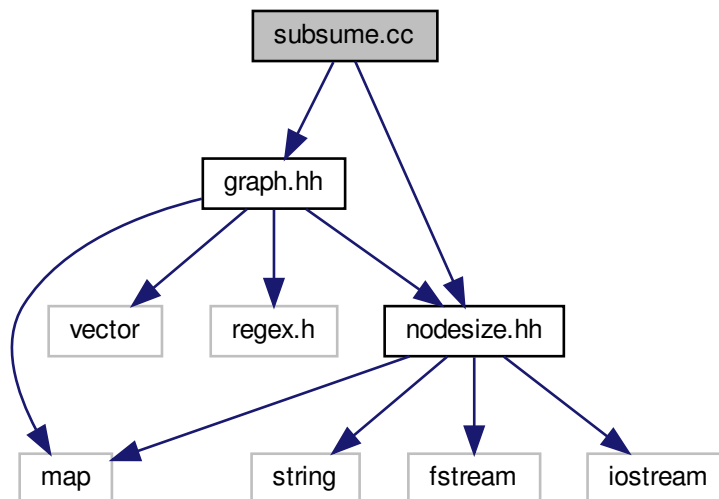
10.20.3.1 scanAnnot()

```
annotmap* scanAnnot (
    string & s,
    void * loc )
```

10.21 README.md File Reference

10.22 subsume.cc File Reference

```
#include <graph.hh>
#include <nodesize.hh>
Include dependency graph for subsume.cc:
```



Macros

- `#define` [ARRAY_SIZE](#)(arr) (sizeof((arr)) / sizeof((arr)[0]))

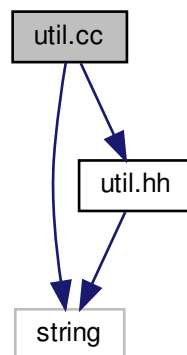
10.22.1 Macro Definition Documentation

10.22.1.1 ARRAY_SIZE

```
#define ARRAY_SIZE(  
    arr ) (sizeof((arr)) / sizeof((arr)[0]))
```

10.23 util.cc File Reference

```
#include <string>  
#include <util.hh>  
Include dependency graph for util.cc:
```



Functions

- string `camelcase` (string s)

10.23.1 Function Documentation

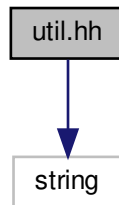
10.23.1.1 camelcase()

```
string camelcase (  
    string s )
```

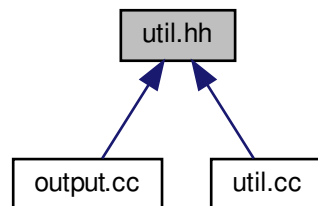

10.24 util.hh File Reference

```
#include <string>
```

Include dependency graph for util.hh:



This graph shows which files directly or indirectly include this file:



Functions

- string [camelcase](#) (string s)

10.24.1 Function Documentation

10.24.1.1 camelcase()

```
string camelcase (  
    string s )
```


Index

- ~basic_symbol
 - annot::parser::basic_symbol< Base >, [28](#)
 - yy::parser::basic_symbol< Base >, [34](#)
- ~choicenode
 - choicenode, [43](#)
- ~concatnode
 - concatnode, [47](#)
- ~grammar
 - grammar, [57](#)
- ~loopnode
 - loopnode, [67](#)
- ~multinode
 - multinode, [71](#)
- ~newlinenode
 - newlinenode, [78](#)
- ~node
 - node, [83](#)
- ~nodesizes
 - nodesizes, [96](#)
- ~nontermnode
 - nontermnode, [99](#)
- ~parser
 - annot::parser, [108](#)
 - yy::parser, [115](#)
- ~productionnode
 - productionnode, [128](#)
- ~railnode
 - railnode, [132](#)
- ~rownode
 - rownode, [137](#)
- ~semantic_type
 - annot::parser::semantic_type, [139](#)
- ~singlenode
 - singlenode, [144](#)
- ~syntax_error
 - annot::parser::syntax_error, [159](#)
 - yy::parser::syntax_error, [161](#)
- ~termnode
 - termnode, [163](#)
- ~value_type
 - yy::parser::value_type, [168](#)
- A
 - lexer.cc, [248](#)
- addString
 - driver, [55](#)
- addTerminal
 - driver, [55](#)
- alloc
 - annot_parser.cc, [233](#)
- analyzeNonOptLoops
 - concatnode, [47](#)
 - multinode, [72](#)
 - node, [83](#)
 - nontermnode, [99](#)
 - singlenode, [144](#)
- analyzeOptLoops
 - concatnode, [47](#)
 - multinode, [72](#)
 - node, [83](#)
 - nontermnode, [99](#)
 - singlenode, [144](#)
- annot, [17](#)
 - operator<<, [19](#), [20](#)
 - operator+, [18](#)
 - operator+=", [18](#)
 - operator-, [19](#)
 - operator=, [19](#)
- annot::location, [59](#)
 - begin, [62](#)
 - columns, [61](#)
 - counter_type, [60](#)
 - end, [62](#)
 - filename_type, [60](#)
 - initialize, [61](#)
 - lines, [61](#)
 - location, [60](#), [61](#)
 - step, [61](#)
- annot::parser, [105](#)
 - ~parser, [108](#)
 - by_type, [107](#)
 - error, [108](#)
 - location_type, [107](#)
 - make_AEND, [109](#)
 - make_ANNOTerror, [109](#)
 - make_ANNOTUNDEF, [109](#)
 - make_AS, [109](#)
 - make_ASTART, [109](#)
 - make_CAPTION, [109](#)
 - make_END, [109](#)
 - make_SEMICOLON, [110](#)
 - make_SIDEWAYS, [110](#)
 - make_STRING, [110](#)
 - make_SUBSUME, [110](#)
 - make_UNEXP, [110](#)
 - operator(), [110](#)
 - parse, [111](#)
 - parser, [108](#)
 - symbol_kind_type, [107](#)

- symbol_name, 111
- token_kind_type, 107
- token_type, 107
- YYNTOKENS, 111
- annot::parser::basic_symbol< Base >, 25
 - ~basic_symbol, 28
 - basic_symbol, 27, 28
 - clear, 29
 - empty, 29
 - location, 30
 - move, 29
 - name, 29
 - super_type, 27
 - type_get, 29
 - value, 30
- annot::parser::by_kind, 36
 - by_kind, 37
 - clear, 37
 - kind, 38
 - kind_, 38
 - kind_type, 37
 - move, 38
 - type_get, 38
- annot::parser::context, 50
 - context, 50
 - expected_tokens, 50
 - location, 50
 - lookahead, 51
 - token, 51
- annot::parser::semantic_type, 138
 - ~semantic_type, 139
 - as, 140
 - build, 140
 - copy, 140
 - destroy, 140
 - emplace, 141
 - move, 141
 - self_type, 139
 - semantic_type, 139
 - swap, 141
 - yyalign_me, 142
 - yyraw, 142
- annot::parser::stack< T, S >::slice, 148
 - operator[], 149
 - slice, 149
- annot::parser::symbol_kind, 150
 - S_AEND, 151
 - S_annot, 151
 - S_annotations, 151
 - S_annots, 151
 - S_AS, 151
 - S_ASTART, 151
 - S_CAPTION, 151
 - S_SEMICOLON, 151
 - S_SIDEWAYS, 151
 - S_STRING, 151
 - S_SUBSUME, 151
 - S_UNEXP, 151
 - S_YYACCEPT, 151
 - S_YYEMPTY, 151
 - S_YYEOF, 151
 - S_YYerror, 151
 - S_YYUNDEF, 151
 - symbol_kind_type, 151
 - YYNTOKENS, 151
- annot::parser::symbol_type, 153
 - super_type, 154
 - symbol_type, 154, 155
- annot::parser::syntax_error, 158
 - ~syntax_error, 159
 - location, 159
 - syntax_error, 159
- annot::parser::token, 164
 - TOK_AEND, 164
 - TOK_ANNOTEMPTY, 164
 - TOK_ANNOTerror, 164
 - TOK_ANNOTUNDEF, 164
 - TOK_AS, 164
 - TOK_ASTART, 164
 - TOK_CAPTION, 164
 - TOK_END, 164
 - TOK_SEMICOLON, 164
 - TOK_SIDEWAYS, 164
 - TOK_STRING, 164
 - TOK_SUBSUME, 164
 - TOK_UNEXP, 164
 - token_kind_type, 164
 - yytokentype, 164
- annot::position, 120
 - column, 122
 - columns, 122
 - counter_type, 121
 - filename, 123
 - filename_type, 121
 - initialize, 122
 - line, 123
 - lines, 122
 - position, 122
- ANNOT_ASSERT
 - annot_parser.hh, 235
- annot_create_buffer_ALREADY_DEFINED
 - annot_lexer.cc, 179
- annot_delete_buffer_ALREADY_DEFINED
 - annot_lexer.cc, 179
- annot_flex_debug_ALREADY_DEFINED
 - annot_lexer.cc, 179
- annot_flush_buffer_ALREADY_DEFINED
 - annot_lexer.cc, 180
- annot_init_buffer_ALREADY_DEFINED
 - annot_lexer.cc, 180
- annot_lexer.cc, 175
 - annot_create_buffer_ALREADY_DEFINED, 179
 - annot_delete_buffer_ALREADY_DEFINED, 179
 - annot_flex_debug_ALREADY_DEFINED, 179
 - annot_flush_buffer_ALREADY_DEFINED, 180
 - annot_init_buffer_ALREADY_DEFINED, 180

annot_load_buffer_state_ALREADY_DEFINED, 180
annot_scan_buffer_ALREADY_DEFINED, 180
annot_scan_bytes_ALREADY_DEFINED, 180
annot_scan_string_ALREADY_DEFINED, 180
annot_switch_to_buffer_ALREADY_DEFINED, 180
annotalloc_ALREADY_DEFINED, 180
annotensure_buffer_stack_ALREADY_DEFINED, 181
annotfree_ALREADY_DEFINED, 181
annotin_ALREADY_DEFINED, 181
annotleng_ALREADY_DEFINED, 181
annotlex_ALREADY_DEFINED, 181
annotlineno_ALREADY_DEFINED, 181
annotout_ALREADY_DEFINED, 181
annotpop_buffer_state_ALREADY_DEFINED, 181
annotpush_buffer_state_ALREADY_DEFINED, 182
annotrealloc_ALREADY_DEFINED, 182
annotrestart_ALREADY_DEFINED, 182
annottext_ALREADY_DEFINED, 182
annotwrap, 182
annotwrap_ALREADY_DEFINED, 182
BEGIN, 182
ECHO, 182
EOB_ACT_CONTINUE_SCAN, 183
EOB_ACT_END_OF_FILE, 183
EOB_ACT_LAST_MATCH, 183
FLEX_BETA, 183
flex_int16_t, 199
flex_int32_t, 199
flex_int8_t, 199
FLEX_SCANNER, 183
flex_uint16_t, 200
flex_uint32_t, 200
flex_uint8_t, 200
FLEXINT_H, 183
if, 201
INITIAL, 183
INT16_MAX, 183
INT16_MIN, 184
INT32_MAX, 184
INT32_MIN, 184
INT8_MAX, 184
INT8_MIN, 184
REJECT, 184
SIZE_MAX, 184
step, 201
stripquotes, 201
UINT16_MAX, 184
UINT32_MAX, 185
UINT8_MAX, 185
unput, 185
while, 201
yy_act, 206
YY_AT_BOL, 185
yy_bp, 206
YY_BREAK, 185
YY_BUF_SIZE, 185
YY_BUFFER_EOF_PENDING, 185
YY_BUFFER_NEW, 186
YY_BUFFER_NORMAL, 186
YY_BUFFER_STATE, 200
YY_CHAR, 200
yy_cp, 206
yy_create_buffer, 186, 201
YY_CURRENT_BUFFER, 186
YY_CURRENT_BUFFER_LVALUE, 186
YY_DECL, 186, 206
yy_delete_buffer, 186, 202
YY_DO_BEFORE_ACTION, 187
YY_END_OF_BUFFER, 187
YY_END_OF_BUFFER_CHAR, 187
YY_EXIT_FAILURE, 187
YY_EXTRA_TYPE, 187
YY_FATAL_ERROR, 187
yy_flex_debug, 187, 206
YY_FLEX_MAJOR_VERSION, 188
YY_FLEX_MINOR_VERSION, 188
YY_FLEX_SUBMINOR_VERSION, 188
YY_FLUSH_BUFFER, 188
yy_flush_buffer, 188, 202
yy_init_buffer, 188
YY_INPUT, 188
YY_INT_ALIGNED, 189
YY_LESS_LINENO, 189
YY_LINENO_REWIND_TO, 189
yy_load_buffer_state, 189
YY_MORE_ADJ, 190
yy_new_buffer, 190
YY_NEW_FILE, 190
YY_NO_INPUT, 190
YY_NULL, 190
YY_NUM_RULES, 190
YY_READ_BUF_SIZE, 190
YY_RESTORE_YY_MORE_OFFSET, 191
YY_RULE_SETUP, 191
YY_SC_TO_UI, 191
yy_scan_buffer, 191, 202
yy_scan_bytes, 191, 203
yy_scan_string, 191, 203
yy_set_bol, 191
yy_set_interactive, 192
yy_size_t, 200
YY_SKIP_YYWRAP, 192
YY_START, 192
YY_START_STACK_INCR, 192
YY_STATE_BUF_SIZE, 192
YY_STATE_EOF, 193
yy_state_type, 200
YY_STRUCT_YY_BUFFER_STATE, 193
yy_switch_to_buffer, 193, 203
YY_TYPEDEF_YY_BUFFER_STATE, 193
YY_TYPEDEF_YY_SIZE_T, 193
YY_USER_ACTION, 193
yyalloc, 193, 204

- yyconst, 194
- yyensure_buffer_stack, 194
- yyfree, 194, 204
- yyget_debug, 194
- yyget_extra, 194
- yyget_in, 194
- yyget_leng, 194
- yyget_lineno, 195
- yyget_out, 195
- yyget_text, 195
- yyin, 195, 207
- yyleng, 195, 207
- yyless, 195, 196
- yylex, 196
- yylex_destroy, 196
- yylex_init, 196
- yylex_init_extra, 196
- yylineno, 197, 207
- yyomore, 197
- yynoreturn, 197
- yyout, 197, 207
- yypop_buffer_state, 197
- yypush_buffer_state, 197, 204
- yyrealloc, 197, 204
- yyrestart, 198, 204
- yyset_debug, 198, 205
- yyset_extra, 198, 205
- yyset_in, 198, 205
- yyset_lineno, 198, 205
- yyset_out, 198, 206
- YYSTATE, 198
- YYTABLES_NAME, 198
- yyterminate, 199
- yytext, 199, 207
- yytext_ptr, 199
- yywrap, 199
- annot_lexer.hh, 208
 - annotIN_HEADER, 210
 - annotwrap, 211
 - FLEX_BETA, 211
 - flex_int16_t, 220
 - flex_int32_t, 220
 - flex_int8_t, 220
 - FLEX_SCANNER, 211
 - flex_uint16_t, 220
 - flex_uint32_t, 220
 - flex_uint8_t, 221
 - FLEXINT_H, 211
 - INT16_MAX, 211
 - INT16_MIN, 211
 - INT32_MAX, 211
 - INT32_MIN, 211
 - INT8_MAX, 212
 - INT8_MIN, 212
 - SIZE_MAX, 212
 - UINT16_MAX, 212
 - UINT32_MAX, 212
 - UINT8_MAX, 212
 - YY_BUF_SIZE, 212
 - YY_BUFFER_STATE, 221
 - yy_create_buffer, 212, 221
 - YY_DECL, 213
 - YY_DECL_IS_OURS, 213
 - yy_delete_buffer, 213, 221
 - YY_EXTRA_TYPE, 213
 - yy_flex_debug, 213
 - YY_FLEX_MAJOR_VERSION, 213
 - YY_FLEX_MINOR_VERSION, 213
 - YY_FLEX_SUBMINOR_VERSION, 213
 - yy_flush_buffer, 214, 222
 - yy_init_buffer, 214
 - YY_INT_ALIGNED, 214
 - yy_load_buffer_state, 214
 - YY_READ_BUF_SIZE, 214
 - yy_scan_buffer, 214, 222
 - yy_scan_bytes, 214, 222
 - yy_scan_string, 215, 223
 - yy_size_t, 221
 - YY_SKIP_YYWRAP, 215
 - YY_START_STACK_INCR, 215
 - YY_STRUCT_YY_BUFFER_STATE, 215
 - yy_switch_to_buffer, 215, 223
 - YY_TYPEDEF_YY_BUFFER_STATE, 215
 - YY_TYPEDEF_YY_SIZE_T, 215
 - yyalloc, 215, 223
 - yyconst, 216
 - yyensure_buffer_stack, 216
 - yyfree, 216, 224
 - yyget_debug, 216
 - yyget_extra, 216
 - yyget_in, 216
 - yyget_leng, 216
 - yyget_lineno, 217
 - yyget_out, 217
 - yyget_text, 217
 - yyin, 217, 226
 - yyleng, 217, 226
 - yylex, 217
 - yylex_destroy, 217
 - yylex_init, 218
 - yylex_init_extra, 218
 - yylineno, 218, 226
 - yynoreturn, 218
 - yyout, 218, 226
 - yypop_buffer_state, 218
 - yypush_buffer_state, 218, 224
 - yyrealloc, 219, 224
 - yyrestart, 219, 224
 - yyset_debug, 219, 225
 - yyset_extra, 219, 225
 - yyset_in, 219, 225
 - yyset_lineno, 219, 225
 - yyset_out, 219, 226
 - yytext, 219, 226
 - yytext_ptr, 220
 - yywrap, 220

- annot_load_buffer_state_ALREADY_DEFINED
 - annot_lexer.cc, [180](#)
- annot_location.hh, [227](#)
 - YY_NULLPTR, [228](#)
- annot_parser.cc, [229](#)
 - alloc, [233](#)
 - loc, [233](#)
 - scanAnnot, [232](#)
 - YY_, [230](#)
 - YY_DECL, [230](#), [233](#)
 - YY_EXCEPTIONS, [230](#)
 - YY_REDUCE_PRINT, [230](#)
 - YY_STACK_PRINT, [230](#)
 - YY_SYMBOL_PRINT, [230](#)
 - YYABORT, [230](#)
 - YYACCEPT, [231](#)
 - YYCASE_, [231](#)
 - YYCDEBUG, [231](#)
 - yyclearin, [231](#)
 - yyerrok, [231](#)
 - YYERROR, [231](#)
 - yylex, [231](#)
 - YYLOC_DEFAULT, [232](#)
 - YYRECOVERING, [232](#)
 - YYRHSLOC, [232](#)
- annot_parser.hh, [233](#)
 - ANNOT_ASSERT, [235](#)
 - ANNOTDEBUG, [235](#)
 - YY_ATTRIBUTE_PURE, [235](#)
 - YY_ATTRIBUTE_UNUSED, [235](#)
 - YY_CAST, [235](#)
 - YY_CONSTEXPR, [236](#)
 - YY_COPY, [236](#)
 - YY_CPLUSPLUS, [236](#)
 - YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN, [236](#)
 - YY_IGNORE_MAYBE_UNINITIALIZED_END, [236](#)
 - YY_IGNORE_USELESS_CAST_BEGIN, [236](#)
 - YY_IGNORE_USELESS_CAST_END, [236](#)
 - YY_INITIAL_VALUE, [237](#)
 - YY_MOVE, [237](#)
 - YY_MOVE_OR_COPY, [237](#)
 - YY_MOVE_REF, [237](#)
 - YY_NOEXCEPT, [237](#)
 - YY_NOTHROW, [237](#)
 - YY_REINTERPRET_CAST, [237](#)
 - YY_RVREF, [238](#)
 - YY_USE, [238](#)
- annot_scan_buffer_ALREADY_DEFINED
 - annot_lexer.cc, [180](#)
- annot_scan_bytes_ALREADY_DEFINED
 - annot_lexer.cc, [180](#)
- annot_scan_string_ALREADY_DEFINED
 - annot_lexer.cc, [180](#)
- annot_switch_to_buffer_ALREADY_DEFINED
 - annot_lexer.cc, [180](#)
- annot_t
 - graph.hh, [243](#)
- annotalloc_ALREADY_DEFINED
 - annot_lexer.cc, [180](#)
- ANNOTDEBUG
 - annot_parser.hh, [235](#)
- annotensure_buffer_stack_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotfree_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotin_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotIN_HEADER
 - annot_lexer.hh, [210](#)
- annotleng_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotlex_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotlineno_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotmap
 - graph.hh, [243](#)
- annotout_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotpop_buffer_state_ALREADY_DEFINED
 - annot_lexer.cc, [181](#)
- annotpush_buffer_state_ALREADY_DEFINED
 - annot_lexer.cc, [182](#)
- annotrealloc_ALREADY_DEFINED
 - annot_lexer.cc, [182](#)
- annotrestart_ALREADY_DEFINED
 - annot_lexer.cc, [182](#)
- annottext_ALREADY_DEFINED
 - annot_lexer.cc, [182](#)
- annotwrap
 - annot_lexer.cc, [182](#)
 - annot_lexer.hh, [211](#)
- annotwrap_ALREADY_DEFINED
 - annot_lexer.cc, [182](#)
- ARRAY_SIZE
 - subsume.cc, [300](#)
- as
 - annot::parser::semantic_type, [140](#)
 - yy::parser::value_type, [168](#)
- basic_symbol
 - annot::parser::basic_symbol< Base >, [27](#), [28](#)
 - yy::parser::basic_symbol< Base >, [32–34](#)
- beforeskip
 - node, [93](#)
- BEGIN
 - annot_lexer.cc, [182](#)
 - lexer.cc, [248](#)
- begin
 - annot::location, [62](#)
 - yy::location, [65](#)
- body
 - singlenode, [148](#)
- bottom
 - railnode, [134](#)
- build

- annot::parser::semantic_type, 140
 - yy::parser::value_type, 168, 169
- by_kind
 - annot::parser::by_kind, 37
 - yy::parser::by_kind, 39, 40
- by_type
 - annot::parser, 107
 - yy::parser, 114
- camelcase
 - util.cc, 300
 - util.hh, 301
- CHOICE
 - node, 83
- choicenode, 41
 - ~choicenode, 43
 - choicenode, 42, 43
 - clone, 43
 - drawToLeftRail, 43
 - drawToRightRail, 43
 - dump, 44
 - fixSkips, 44
 - insert, 44
 - mergeChoices, 44
 - rail_left, 44
 - rail_right, 45
 - texName, 45
- clear
 - annot::parser::basic_symbol< Base >, 29
 - annot::parser::by_kind, 37
 - yy::parser::basic_symbol< Base >, 34
 - yy::parser::by_kind, 40
- clone
 - choicenode, 43
 - concatnode, 47
 - loopnode, 67
 - multinode, 72
 - newlinenode, 78
 - node, 84
 - nontermnode, 99
 - nullnode, 105
 - productionnode, 128
 - railnode, 132
 - rownode, 137
 - singlenode, 144
 - termnode, 163
- CODE_OF_CONDUCT.md, 238
- colsep
 - nodesizes, 96
- column
 - annot::position, 122
 - yy::position, 125
- columns
 - annot::location, 61
 - annot::position, 122
 - yy::location, 64
 - yy::position, 125
- CONCAT
 - node, 83
- concatnode, 45
 - ~concatnode, 47
 - analyzeNonOptLoops, 47
 - analyzeOptLoops, 47
 - clone, 47
 - concatnode, 46, 47
 - createRows, 47
 - drawToLeftRail, 48
 - drawToRightRail, 48
 - dump, 48
 - fixSkips, 48
 - insert, 48
 - mergeConcats, 49
 - mergeRails, 49
 - multinode, 76
 - place, 49
 - setNext, 49
 - setPrevious, 49
- context
 - annot::parser::context, 50
 - yy::parser::context, 51
- CONTRIBUTING.md, 238
- coordinate, 52
 - coordinate, 53
 - operator<<, 54
 - operator+, 53
 - operator-, 53
 - operator=, 53
 - x, 54
 - y, 54
- copy
 - annot::parser::semantic_type, 140
 - yy::parser::value_type, 169
- counter_type
 - annot::location, 60
 - annot::position, 121
 - yy::location, 63
 - yy::position, 124
- createRows
 - concatnode, 47
 - grammar, 57
 - node, 84
 - productionnode, 128
- dead
 - node, 93
- debug_level
 - yy::parser, 115
- debug_level_type
 - yy::parser, 114
- debug_stream
 - yy::parser, 115
- deleteData
 - node, 84
- description
 - main.cc, 287
- destroy
 - annot::parser::semantic_type, 140
 - yy::parser::value_type, 169

- direction
 - railnode, [134](#)
- DOWN
 - graph.hh, [244](#)
- drawToLeftRail
 - choicenode, [43](#)
 - concatnode, [48](#)
 - loopnode, [68](#)
 - node, [84](#)
 - nontermnode, [100](#)
 - singlenode, [145](#)
- drawtoprev
 - node, [93](#)
- drawToRightRail
 - choicenode, [43](#)
 - concatnode, [48](#)
 - loopnode, [68](#)
 - node, [84](#)
 - nontermnode, [100](#)
 - singlenode, [145](#)
- driver, [54](#)
 - addString, [55](#)
 - addTerminal, [55](#)
 - driver, [55](#)
 - get_location, [55](#)
 - get_result, [55](#)
 - outs, [55](#)
 - parse, [55](#)
 - scan_begin, [56](#)
 - scan_end, [56](#)
- driver.cc, [238](#)
- driver.hh, [239](#)
 - YY_DECL, [240](#)
- dump
 - choicenode, [44](#)
 - concatnode, [48](#)
 - grammar, [57](#)
 - loopnode, [68](#)
 - newlinenode, [78](#)
 - node, [85](#)
 - nontermnode, [100](#)
 - productionnode, [128](#)
 - railnode, [132](#)
 - rownode, [137](#)
- ea
 - node, [93](#)
- east
 - node, [85](#)
- ECHO
 - annot_lexer.cc, [182](#)
 - lexer.cc, [248](#)
- emplace
 - annot::parser::semantic_type, [141](#)
 - yy::parser::value_type, [169](#)
- empty
 - annot::parser::basic_symbol< Base >, [29](#)
 - yy::parser::basic_symbol< Base >, [34](#)
- end
 - annot::location, [62](#)
 - yy::location, [65](#)
- EOB_ACT_CONTINUE_SCAN
 - annot_lexer.cc, [183](#)
 - lexer.cc, [249](#)
- EOB_ACT_END_OF_FILE
 - annot_lexer.cc, [183](#)
 - lexer.cc, [249](#)
- EOB_ACT_LAST_MATCH
 - annot_lexer.cc, [183](#)
 - lexer.cc, [249](#)
- error
 - annot::parser, [108](#)
 - yy::parser, [115](#), [116](#)
- expected_tokens
 - annot::parser::context, [50](#)
 - yy::parser::context, [52](#)
- filename
 - annot::position, [123](#)
 - yy::position, [125](#)
- filename_type
 - annot::location, [60](#)
 - annot::position, [121](#)
 - yy::location, [63](#)
 - yy::position, [124](#)
- fixSkips
 - choicenode, [44](#)
 - concatnode, [48](#)
 - grammar, [57](#)
 - loopnode, [68](#)
 - multinode, [72](#)
 - node, [85](#)
 - productionnode, [128](#)
 - singlenode, [145](#)
- FLEX_BETA
 - annot_lexer.cc, [183](#)
 - annot_lexer.hh, [211](#)
 - lexer.cc, [249](#)
 - lexer.hh, [271](#)
- FLEX_DEBUG
 - lexer.cc, [249](#)
 - lexer.hh, [271](#)
- flex_int16_t
 - annot_lexer.cc, [199](#)
 - annot_lexer.hh, [220](#)
 - lexer.cc, [259](#)
 - lexer.hh, [276](#)
- flex_int32_t
 - annot_lexer.cc, [199](#)
 - annot_lexer.hh, [220](#)
 - lexer.cc, [259](#)
 - lexer.hh, [276](#)
- flex_int8_t
 - annot_lexer.cc, [199](#)
 - annot_lexer.hh, [220](#)
 - lexer.cc, [259](#)
 - lexer.hh, [276](#)
- FLEX_SCANNER

- annot_lexer.cc, [183](#)
 - annot_lexer.hh, [211](#)
 - lexer.cc, [249](#)
 - lexer.hh, [272](#)
- flex_uint16_t
 - annot_lexer.cc, [200](#)
 - annot_lexer.hh, [220](#)
 - lexer.cc, [260](#)
 - lexer.hh, [276](#)
- flex_uint32_t
 - annot_lexer.cc, [200](#)
 - annot_lexer.hh, [220](#)
 - lexer.cc, [260](#)
 - lexer.hh, [276](#)
- flex_uint8_t
 - annot_lexer.cc, [200](#)
 - annot_lexer.hh, [221](#)
 - lexer.cc, [260](#)
 - lexer.hh, [276](#)
- FLEXINT_H
 - annot_lexer.cc, [183](#)
 - annot_lexer.hh, [211](#)
 - lexer.cc, [249](#)
 - lexer.hh, [272](#)
- forgetChild
 - multinode, [72](#)
 - node, [85](#)
 - nontermnode, [100](#)
 - singlenode, [145](#)
- format
 - nontermnode, [102](#)
- get_location
 - driver, [55](#)
- get_result
 - driver, [55](#)
- getBeforeSkip
 - node, [85](#)
- getBody
 - loopnode, [68](#)
- getBottom
 - railnode, [132](#)
- getChild
 - multinode, [73](#)
 - node, [85](#)
 - nontermnode, [100](#)
 - singlenode, [145](#)
- getColSep
 - node, [86](#)
- getDrawToPrev
 - node, [86](#)
- getLeftRail
 - node, [86](#)
- getName
 - productionnode, [129](#)
- getNext
 - node, [86](#)
- getParent
 - node, [86](#)
- getPrevious
 - node, [86](#)
- getRailDir
 - railnode, [133](#)
- getRepeat
 - loopnode, [69](#)
- getRightRail
 - node, [86](#)
- getSize
 - nodesizes, [96](#)
- getSubsume
 - productionnode, [129](#)
- GRAMMAR
 - node, [83](#)
- grammar, [56](#)
 - ~grammar, [57](#)
 - createRows, [57](#)
 - dump, [57](#)
 - fixSkips, [57](#)
 - grammar, [57](#)
 - insert, [57](#)
 - mergeRails, [57](#)
 - optimize, [58](#)
 - place, [58](#)
 - setNext, [58](#)
 - setParent, [58](#)
 - setPrevious, [58](#)
 - subsume, [58](#)
- graph.cc, [240](#)
 - latexwrite, [241](#)
 - nextChain, [241](#)
 - nextCoord, [241](#)
 - nextFit, [241](#)
 - nextNode, [241](#)
 - stripSpecial, [241](#)
- graph.hh, [242](#)
 - annot_t, [243](#)
 - annotmap, [243](#)
 - DOWN, [244](#)
 - latexwrite, [244](#)
 - LEFT, [244](#)
 - nextChain, [244](#)
 - nextCoord, [244](#)
 - nextFit, [244](#)
 - nextNode, [244](#)
 - RIGHT, [244](#)
 - stripSpecial, [245](#)
 - UP, [244](#)
 - vraildir, [243](#)
 - vrailside, [244](#)
- height
 - node, [87](#)
- if
 - annot_lexer.cc, [201](#)
 - lexer.cc, [261](#)
- INITIAL
 - annot_lexer.cc, [183](#)

- lexer.cc, 249
- initialize
 - annot::location, 61
 - annot::position, 122
 - yy::location, 64
 - yy::position, 125
- insert
 - choicenode, 44
 - concatnode, 48
 - grammar, 57
 - multinode, 73
 - node, 87
- insertFirst
 - multinode, 73
- INT16_MAX
 - annot_lexer.cc, 183
 - annot_lexer.hh, 211
 - lexer.cc, 250
 - lexer.hh, 272
- INT16_MIN
 - annot_lexer.cc, 184
 - annot_lexer.hh, 211
 - lexer.cc, 250
 - lexer.hh, 272
- INT32_MAX
 - annot_lexer.cc, 184
 - annot_lexer.hh, 211
 - lexer.cc, 250
 - lexer.hh, 272
- INT32_MIN
 - annot_lexer.cc, 184
 - annot_lexer.hh, 211
 - lexer.cc, 250
 - lexer.hh, 272
- INT8_MAX
 - annot_lexer.cc, 184
 - annot_lexer.hh, 212
 - lexer.cc, 250
 - lexer.hh, 272
- INT8_MIN
 - annot_lexer.cc, 184
 - annot_lexer.hh, 212
 - lexer.cc, 250
 - lexer.hh, 272
- is_choice
 - node, 87
- is_concat
 - node, 87
- is_loop
 - node, 87
- is_newline
 - node, 87
- is_nonterm
 - node, 87
- is_null
 - node, 88
- is_production
 - node, 88
- is_rail
 - node, 88
- is_row
 - node, 88
- is_terminal
 - node, 88
- isDead
 - node, 88
- kind
 - annot::parser::by_kind, 38
 - yy::parser::by_kind, 40
- kind_
 - annot::parser::by_kind, 38
 - yy::parser::by_kind, 41
- kind_type
 - annot::parser::by_kind, 37
 - yy::parser::by_kind, 39
- latexwrite
 - graph.cc, 241
 - graph.hh, 244
- LEFT
 - graph.hh, 244
- leftrail
 - node, 93
- lexer.cc, 245
 - A, 248
 - BEGIN, 248
 - ECHO, 248
 - EOB_ACT_CONTINUE_SCAN, 249
 - EOB_ACT_END_OF_FILE, 249
 - EOB_ACT_LAST_MATCH, 249
 - FLEX_BETA, 249
 - FLEX_DEBUG, 249
 - flex_int16_t, 259
 - flex_int32_t, 259
 - flex_int8_t, 259
 - FLEX_SCANNER, 249
 - flex_uint16_t, 260
 - flex_uint32_t, 260
 - flex_uint8_t, 260
 - FLEXINT_H, 249
 - if, 261
 - INITIAL, 249
 - INT16_MAX, 250
 - INT16_MIN, 250
 - INT32_MAX, 250
 - INT32_MIN, 250
 - INT8_MAX, 250
 - INT8_MIN, 250
 - REJECT, 250
 - SIZE_MAX, 250
 - step, 261
 - subloc, 267
 - UINT16_MAX, 251
 - UINT32_MAX, 251
 - UINT8_MAX, 251
 - unput, 251

- while, 261
- yy_act, 267
- YY_AT_BOL, 251
- yy_bp, 267
- YY_BREAK, 251
- YY_BUF_SIZE, 251
- YY_BUFFER_EOF_PENDING, 252
- YY_BUFFER_NEW, 252
- YY_BUFFER_NORMAL, 252
- YY_BUFFER_STATE, 260
- YY_CHAR, 260
- yy_cp, 267
- yy_create_buffer, 261
- YY_CURRENT_BUFFER, 252
- YY_CURRENT_BUFFER_LVALUE, 252
- YY_DECL, 268
- yy_delete_buffer, 261
- YY_DO_BEFORE_ACTION, 252
- YY_END_OF_BUFFER, 252
- YY_END_OF_BUFFER_CHAR, 253
- YY_EXIT_FAILURE, 253
- YY_EXTRA_TYPE, 253
- YY_FATAL_ERROR, 253
- yy_flex_debug, 268
- YY_FLEX_MAJOR_VERSION, 253
- YY_FLEX_MINOR_VERSION, 253
- YY_FLEX_SUBMINOR_VERSION, 253
- YY_FLUSH_BUFFER, 254
- yy_flush_buffer, 262
- YY_INPUT, 254
- YY_INT_ALIGNED, 254
- YY_LESS_LINENO, 254
- YY_LINENO_REWIND_TO, 254
- YY_MORE_ADJ, 254
- yy_new_buffer, 254
- YY_NEW_FILE, 255
- YY_NO_INPUT, 255
- YY_NULL, 255
- YY_NUM_RULES, 255
- YY_READ_BUF_SIZE, 255
- YY_RESTORE YY_MORE_OFFSET, 255
- YY_RULE_SETUP, 255
- YY_SC_TO_UI, 255
- yy_scan_buffer, 262
- yy_scan_bytes, 262
- yy_scan_string, 263
- yy_set_bol, 256
- yy_set_interactive, 256
- yy_size_t, 260
- YY_SKIP YYWRAP, 256
- YY_START, 256
- YY_START_STACK_INCR, 256
- YY_STATE_BUF_SIZE, 257
- YY_STATE_EOF, 257
- yy_state_type, 260
- YY_STRUCT YY_BUFFER_STATE, 257
- yy_switch_to_buffer, 263
- YY_TYPEDEF YY_BUFFER_STATE, 257
- YY_TYPEDEF YY_SIZE_T, 257
- YY_USER_ACTION, 257
- yyalloc, 263
- yyconst, 257
- yyfree, 264
- yyget_debug, 264
- yyget_extra, 264
- yyget_in, 264
- yyget_leng, 264
- yyget_lineno, 264
- yyget_out, 264
- yyget_text, 265
- yyin, 268
- yylen, 268
- yyless, 258
- yylex_destroy, 265
- yylineno, 268
- yymore, 258
- yyreturn, 258
- yyout, 268
- yypop_buffer_state, 265
- yypush_buffer_state, 265
- yyrealloc, 265
- yyrestart, 265
- yyset_debug, 266
- yyset_extra, 266
- yyset_in, 266
- yyset_lineno, 266
- yyset_out, 267
- YYSTATE, 258
- YYTABLES_NAME, 259
- yyterminate, 259
- yytext, 268
- yytext_ptr, 259
- yywrap, 259
- lexer.hh, 269
- FLEX_BETA, 271
- FLEX_DEBUG, 271
- flex_int16_t, 276
- flex_int32_t, 276
- flex_int8_t, 276
- FLEX_SCANNER, 272
- flex_uint16_t, 276
- flex_uint32_t, 276
- flex_uint8_t, 276
- FLEXINT_H, 272
- INT16_MAX, 272
- INT16_MIN, 272
- INT32_MAX, 272
- INT32_MIN, 272
- INT8_MAX, 272
- INT8_MIN, 272
- SIZE_MAX, 273
- UINT16_MAX, 273
- UINT32_MAX, 273
- UINT8_MAX, 273
- YY_BUF_SIZE, 273
- YY_BUFFER_STATE, 276

- yy_create_buffer, 277
- YY_DECL, 273
- YY_DECL_IS_OURS, 273
- yy_delete_buffer, 277
- YY_EXTRA_TYPE, 273
- YY_FLEX_MAJOR_VERSION, 274
- YY_FLEX_MINOR_VERSION, 274
- YY_FLEX_SUBMINOR_VERSION, 274
- yy_flush_buffer, 277
- YY_INT_ALIGNED, 274
- YY_READ_BUF_SIZE, 274
- yy_scan_buffer, 278
- yy_scan_bytes, 278
- yy_scan_string, 278
- yy_size_t, 276
- YY_SKIP_YYWRAP, 274
- YY_START_STACK_INCR, 274
- YY_STRUCT_YY_BUFFER_STATE, 274
- yy_switch_to_buffer, 279
- YY_TYPEDEF_YY_BUFFER_STATE, 275
- YY_TYPEDEF_YY_SIZE_T, 275
- yyalloc, 279
- yyconst, 275
- yyfree, 279
- yyget_debug, 279
- yyget_extra, 279
- yyget_in, 280
- yyget_leng, 280
- yyget_lineno, 280
- yyget_out, 280
- yyget_text, 280
- yyin, 283
- yyIN_HEADER, 275
- yyleng, 283
- yylex, 280
- yylex_destroy, 280
- yylineno, 283
- yynoreturn, 275
- yyout, 283
- yyvspop_buffer_state, 281
- yyvspush_buffer_state, 281
- yyrealloc, 281
- yyrestart, 281
- yyset_debug, 282
- yyset_extra, 282
- yyset_in, 282
- yyset_lineno, 282
- yyset_out, 283
- yytext, 283
- yytext_ptr, 275
- yywrap, 275
- liftConcats
 - multinode, 73
 - node, 88
 - nontermnode, 101
 - singlenode, 146
- line
 - annot::position, 123
- node, 89
- yy::position, 125
- lines
 - annot::location, 61
 - annot::position, 122
 - yy::location, 65
 - yy::position, 125
- loadData
 - node, 89
 - nodesizes, 96
- loc
 - annot_parser.cc, 233
 - parser.cc, 293
- location
 - annot::location, 60, 61
 - annot::parser::basic_symbol< Base >, 30
 - annot::parser::context, 50
 - annot::parser::syntax_error, 159
 - node, 94
 - yy::location, 64
 - yy::parser::basic_symbol< Base >, 35
 - yy::parser::context, 52
 - yy::parser::syntax_error, 161
- location.hh, 284
- YY_NULLPTR, 285
- location_type
 - annot::parser, 107
 - yy::parser, 114
- lookahead
 - annot::parser::context, 51
 - yy::parser::context, 52
- LOOP
 - node, 83
- loopnode, 66
 - ~loopnode, 67
 - clone, 67
 - drawToLeftRail, 68
 - drawToRightRail, 68
 - dump, 68
 - fixSkips, 68
 - getBody, 68
 - getRepeat, 69
 - loopnode, 67
 - setBody, 69
 - setRepeat, 69
 - texName, 69
- main
 - main.cc, 286
- main.cc, 286
 - description, 287
 - main, 286
 - options, 287
 - optstring, 287
 - usage, 286
- make_AEND
 - annot::parser, 109
- make_ANNOTATION
 - yy::parser, 116

- make_ANNOTerror
 - annot::parser, 109
- make_ANNOTUNDEF
 - annot::parser, 109
- make_AS
 - annot::parser, 109
- make_ASTART
 - annot::parser, 109
- make_CAPTION
 - annot::parser, 109
- make_COMMA
 - yy::parser, 116
- make_END
 - annot::parser, 109
 - yy::parser, 116
- make_EQUAL
 - yy::parser, 116
- make_LBRACE
 - yy::parser, 116
- make_LBRACK
 - yy::parser, 117
- make_LPAREN
 - yy::parser, 117
- make_NEWLINE
 - yy::parser, 117
- make_PIPE
 - yy::parser, 117
- make_RBRACE
 - yy::parser, 117
- make_RBRACK
 - yy::parser, 117
- make_RPAREN
 - yy::parser, 117
- make_SEMICOLON
 - annot::parser, 110
 - yy::parser, 118
- make_SIDEWAYS
 - annot::parser, 110
- make_STRING
 - annot::parser, 110
 - yy::parser, 118
- make_SUBSUME
 - annot::parser, 110
- make_TERM
 - yy::parser, 118
- make_UNEXP
 - annot::parser, 110
 - yy::parser, 118
- make_YError
 - yy::parser, 118
- make_YYUNDEF
 - yy::parser, 118
- makeDead
 - node, 89
- mergeChoices
 - choicenode, 44
 - multinode, 73
 - node, 89
- nontermnode, 101
- singlenode, 146
- mergeConcats
 - concatnode, 49
 - multinode, 74
 - node, 89
 - nontermnode, 101
 - singlenode, 146
- mergeRails
 - concatnode, 49
 - grammar, 57
 - multinode, 74
 - node, 89
 - singlenode, 146
- minsize
 - nodesizes, 96
- move
 - annot::parser::basic_symbol< Base >, 29
 - annot::parser::by_kind, 38
 - annot::parser::semantic_type, 141
 - yy::parser::basic_symbol< Base >, 34
 - yy::parser::by_kind, 40
 - yy::parser::value_type, 169
- multinode, 70
 - ~multinode, 71
 - analyzeNonOptLoops, 72
 - analyzeOptLoops, 72
 - clone, 72
 - concatnode, 76
 - fixSkips, 72
 - forgetChild, 72
 - getChild, 73
 - insert, 73
 - insertFirst, 73
 - liftConcats, 73
 - mergeChoices, 73
 - mergeConcats, 74
 - mergeRails, 74
 - multinode, 71
 - nodes, 76
 - numChildren, 74
 - operator!=, 74
 - operator==, 74
 - place, 75
 - setNext, 75
 - setParent, 75
 - setPrevious, 75
 - subsume, 75
 - texName, 76
- myHeight
 - node, 94
- myWidth
 - node, 94
- name
 - annot::parser::basic_symbol< Base >, 29
 - yy::parser::basic_symbol< Base >, 35
- NEWLINE
 - node, 83

- newlinenode, 77
 - ~newlinenode, 78
 - clone, 78
 - dump, 78
 - newlinenode, 78
 - place, 79
 - rail_left, 79
 - rail_right, 79
 - setLineHeight, 79
- next
 - node, 94
- nextChain
 - graph.cc, 241
 - graph.hh, 244
- nextCoord
 - graph.cc, 241
 - graph.hh, 244
- nextFit
 - graph.cc, 241
 - graph.hh, 244
- nextNode
 - graph.cc, 241
 - graph.hh, 244
- node, 80
 - ~node, 83
 - analyzeNonOptLoops, 83
 - analyzeOptLoops, 83
 - beforeSkip, 93
 - CHOICE, 83
 - clone, 84
 - CONCAT, 83
 - createRows, 84
 - dead, 93
 - deleteData, 84
 - drawToLeftRail, 84
 - drawtoprev, 93
 - drawToRightRail, 84
 - dump, 85
 - ea, 93
 - east, 85
 - fixSkips, 85
 - forgetChild, 85
 - getBeforeSkip, 85
 - getChild, 85
 - getColSep, 86
 - getDrawToPrev, 86
 - getLeftRail, 86
 - getNext, 86
 - getParent, 86
 - getPrevious, 86
 - getRightRail, 86
 - GRAMMAR, 83
 - height, 87
 - insert, 87
 - is_choice, 87
 - is_concat, 87
 - is_loop, 87
 - is_newline, 87
 - is_nonterm, 87
 - is_null, 88
 - is_production, 88
 - is_rail, 88
 - is_row, 88
 - is_terminal, 88
 - isDead, 88
 - lefttrail, 93
 - liftConcats, 88
 - line, 89
 - loadData, 89
 - location, 94
 - LOOP, 83
 - makeDead, 89
 - mergeChoices, 89
 - mergeConcats, 89
 - mergeRails, 89
 - myHeight, 94
 - myWidth, 94
 - NEWLINE, 83
 - next, 94
 - node, 83
 - nodename, 94
 - nodetype, 82
 - NONTERM, 83
 - NULLNODE, 83
 - numChildren, 90
 - operator!=, 90
 - operator==, 90
 - parent, 94
 - place, 90
 - previous, 94
 - PRODUCTION, 83
 - RAIL, 83
 - rawName, 90
 - righttrail, 94
 - ROW, 83
 - same_type, 91
 - setBeforeSkip, 91
 - setDrawToPrev, 91
 - setheight, 91
 - setLeftRail, 91
 - setNext, 91
 - setParent, 91
 - setPrevious, 92
 - setRightRail, 92
 - setWidth, 92
 - sizes, 95
 - subsume, 92
 - TERMINAL, 83
 - texName, 92
 - type, 95
 - UNKNOWN, 83
 - vrailStr, 92
 - wa, 95
 - west, 93
 - width, 93
- nodename

- node, [94](#)
- nodes
 - multinode, [76](#)
- nodesize.hh, [288](#)
- nodesizes, [95](#)
 - ~nodesizes, [96](#)
 - colsep, [96](#)
 - getSize, [96](#)
 - loadData, [96](#)
 - minsize, [96](#)
 - nodesizes, [96](#)
 - rowsep, [97](#)
- nodetype
 - node, [82](#)
- NONTERM
 - node, [83](#)
- nontermnode, [97](#)
 - ~nontermnode, [99](#)
 - analyzeNonOptLoops, [99](#)
 - analyzeOptLoops, [99](#)
 - clone, [99](#)
 - drawToLeftRail, [100](#)
 - drawToRightRail, [100](#)
 - dump, [100](#)
 - forgetChild, [100](#)
 - format, [102](#)
 - getChild, [100](#)
 - liftConcats, [101](#)
 - mergeChoices, [101](#)
 - mergeConcats, [101](#)
 - nontermnode, [99](#)
 - operator!=, [101](#)
 - operator==, [101](#)
 - place, [101](#)
 - str, [102](#)
 - style, [102](#)
 - subsume, [102](#)
 - texName, [102](#)
- NULLNODE
 - node, [83](#)
- nullnode, [103](#)
 - clone, [105](#)
 - nullnode, [104](#)
 - place, [105](#)
 - texName, [105](#)
- numChildren
 - multinode, [74](#)
 - node, [90](#)
 - singlenode, [146](#)
- operator!=
 - multinode, [74](#)
 - node, [90](#)
 - nontermnode, [101](#)
 - railnode, [133](#)
 - singlenode, [146](#)
- operator<<
 - annot, [19](#), [20](#)
 - coordinate, [54](#)
 - yy, [23](#)
- operator()
 - annot::parser, [110](#)
 - yy::parser, [119](#)
- operator+
 - annot, [18](#)
 - coordinate, [53](#)
 - yy, [21](#)
- operator+=
 - annot, [18](#)
 - yy, [21](#), [22](#)
- operator-
 - annot, [19](#)
 - coordinate, [53](#)
 - yy, [22](#)
- operator-=
 - annot, [19](#)
 - yy, [22](#), [23](#)
- operator=
 - coordinate, [53](#)
- operator==
 - multinode, [74](#)
 - node, [90](#)
 - nontermnode, [101](#)
 - railnode, [133](#)
 - singlenode, [147](#)
- operator[]
 - annot::parser::stack< T, S >::slice, [149](#)
 - yy::parser::stack< T, S >::slice, [150](#)
- optimize
 - grammar, [58](#)
 - productionnode, [129](#)
- optimize.cc, [288](#)
- options
 - main.cc, [287](#)
- optstring
 - main.cc, [287](#)
- output.cc, [289](#)
- outs
 - driver, [55](#)
- parent
 - node, [94](#)
- parse
 - annot::parser, [111](#)
 - driver, [55](#)
 - yy::parser, [119](#)
- parser
 - annot::parser, [108](#)
 - yy::parser, [115](#)
- parser.cc, [290](#)
 - loc, [293](#)
 - wrapChoice, [293](#)
 - YY_, [291](#)
 - YY_EXCEPTIONS, [291](#)
 - YY_REDUCE_PRINT, [291](#)
 - YY_STACK_PRINT, [291](#)
 - YY_SYMBOL_PRINT, [291](#)
 - YYABORT, [291](#)

- YYACCEPT, [292](#)
- YYCASE_, [292](#)
- YYCDEBUG, [292](#)
- yyclearin, [292](#)
- yyerrok, [292](#)
- YYERROR, [292](#)
- YYLLOC_DEFAULT, [292](#)
- YYRECOVERING, [293](#)
- YYRHSLOC, [293](#)
- parser.hh, [294](#)
 - scanAnnot, [299](#)
 - YY_ASSERT, [296](#)
 - YY_ATTRIBUTE_PURE, [296](#)
 - YY_ATTRIBUTE_UNUSED, [296](#)
 - YY_CAST, [296](#)
 - YY_CONSTEXPR, [296](#)
 - YY_COPY, [296](#)
 - YY_CPLUSPLUS, [296](#)
 - YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN, [297](#)
 - YY_IGNORE_MAYBE_UNINITIALIZED_END, [297](#)
 - YY_IGNORE_USELESS_CAST_BEGIN, [297](#)
 - YY_IGNORE_USELESS_CAST_END, [297](#)
 - YY_INITIAL_VALUE, [297](#)
 - YY_MOVE, [297](#)
 - YY_MOVE_OR_COPY, [297](#)
 - YY_MOVE_REF, [298](#)
 - YY_NOEXCEPT, [298](#)
 - YY_NOTHROW, [298](#)
 - YY_REINTERPRET_CAST, [298](#)
 - YY_RVREF, [298](#)
 - YY_USE, [298](#)
 - YYDEBUG, [298](#)
- place
 - concatnode, [49](#)
 - grammar, [58](#)
 - multinode, [75](#)
 - newlinenode, [79](#)
 - node, [90](#)
 - nontermnode, [101](#)
 - nullnode, [105](#)
 - productionnode, [129](#)
 - railnode, [133](#)
 - rownode, [137](#)
- position
 - annot::position, [122](#)
 - yy::position, [124](#)
- previous
 - node, [94](#)
- PRODUCTION
 - node, [83](#)
- productionnode, [126](#)
 - ~productionnode, [128](#)
 - clone, [128](#)
 - createRows, [128](#)
 - dump, [128](#)
 - fixSkips, [128](#)
 - getName, [129](#)
 - getSubsume, [129](#)
 - optimize, [129](#)
 - place, [129](#)
 - productionnode, [127](#), [128](#)
 - subsume, [129](#)
 - texName, [129](#)
- RAIL
 - node, [83](#)
- rail_left
 - choicenode, [44](#)
 - newlinenode, [79](#)
- rail_right
 - choicenode, [45](#)
 - newlinenode, [79](#)
- railnode, [130](#)
 - ~railnode, [132](#)
 - bottom, [134](#)
 - clone, [132](#)
 - direction, [134](#)
 - dump, [132](#)
 - getBottom, [132](#)
 - getRailDir, [133](#)
 - operator!=, [133](#)
 - operator==, [133](#)
 - place, [133](#)
 - railnode, [131](#), [132](#)
 - setBottom, [133](#)
 - setRailDir, [134](#)
 - side, [134](#)
 - texName, [134](#)
 - top, [134](#)
- rawName
 - node, [90](#)
- README.md, [299](#)
- REJECT
 - annot_lexer.cc, [184](#)
 - lexer.cc, [250](#)
- RIGHT
 - graph.hh, [244](#)
- rightrail
 - node, [94](#)
- ROW
 - node, [83](#)
- rownode, [135](#)
 - ~rownode, [137](#)
 - clone, [137](#)
 - dump, [137](#)
 - place, [137](#)
 - rownode, [136](#)
 - texName, [137](#)
- rowsep
 - nodesizes, [97](#)
- S_AEND
 - annot::parser::symbol_kind, [151](#)
- S_annot
 - annot::parser::symbol_kind, [151](#)
- S_ANNOTATION

- yy::parser::symbol_kind, [152](#)
- S_annotations
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- S_annots
 - annot::parser::symbol_kind, [151](#)
- S_AS
 - annot::parser::symbol_kind, [151](#)
- S_ASTART
 - annot::parser::symbol_kind, [151](#)
- S_CAPTION
 - annot::parser::symbol_kind, [151](#)
- S_COMMA
 - yy::parser::symbol_kind, [152](#)
- S_EQUAL
 - yy::parser::symbol_kind, [152](#)
- S_expression
 - yy::parser::symbol_kind, [152](#)
- S_grammar
 - yy::parser::symbol_kind, [152](#)
- S_LBRACE
 - yy::parser::symbol_kind, [152](#)
- S_LBRACK
 - yy::parser::symbol_kind, [152](#)
- S_LPAREN
 - yy::parser::symbol_kind, [152](#)
- S_NEWLINE
 - yy::parser::symbol_kind, [152](#)
- S_PIPE
 - yy::parser::symbol_kind, [152](#)
- S_primary
 - yy::parser::symbol_kind, [152](#)
- S_production
 - yy::parser::symbol_kind, [152](#)
- S_productions
 - yy::parser::symbol_kind, [152](#)
- S_RBRACE
 - yy::parser::symbol_kind, [152](#)
- S_RBRACK
 - yy::parser::symbol_kind, [152](#)
- S_rows
 - yy::parser::symbol_kind, [152](#)
- S_RPAREN
 - yy::parser::symbol_kind, [152](#)
- S_SEMICOLON
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- S_SIDEWAYS
 - annot::parser::symbol_kind, [151](#)
- S_STRING
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- S_SUBSUME
 - annot::parser::symbol_kind, [151](#)
- S_TERM
 - yy::parser::symbol_kind, [152](#)
- S_UNEXP
 - annot::parser::symbol_kind, [151](#)
- yy::parser::symbol_kind, [152](#)
- S_YYACCEPT
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- S_YYEMPTY
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- S_YYEOF
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- S_YYerror
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- S_YYUNDEF
 - annot::parser::symbol_kind, [151](#)
 - yy::parser::symbol_kind, [152](#)
- same_type
 - node, [91](#)
- scan_begin
 - driver, [56](#)
- scan_end
 - driver, [56](#)
- scanAnnot
 - annot_parser.cc, [232](#)
 - parser.hh, [299](#)
- self_type
 - annot::parser::semantic_type, [139](#)
 - yy::parser::value_type, [167](#)
- semantic_type
 - annot::parser::semantic_type, [139](#)
 - yy::parser, [114](#)
- set_debug_level
 - yy::parser, [119](#)
- set_debug_stream
 - yy::parser, [119](#)
- setBeforeSkip
 - node, [91](#)
- setBody
 - loopnode, [69](#)
- setBottom
 - railnode, [133](#)
- setDrawToPrev
 - node, [91](#)
- setheight
 - node, [91](#)
- setLeftRail
 - node, [91](#)
- setLineHeight
 - newlinenode, [79](#)
- setNext
 - concatnode, [49](#)
 - grammar, [58](#)
 - multinode, [75](#)
 - node, [91](#)
 - singlenode, [147](#)
- setParent
 - grammar, [58](#)
 - multinode, [75](#)

- node, [91](#)
- singlenode, [147](#)
- setPrevious
 - concatnode, [49](#)
 - grammar, [58](#)
 - multinode, [75](#)
 - node, [92](#)
 - singlenode, [147](#)
- setRailDir
 - railnode, [134](#)
- setRepeat
 - loopnode, [69](#)
- setRightRail
 - node, [92](#)
- setwidth
 - node, [92](#)
- side
 - railnode, [134](#)
- singlenode, [142](#)
 - ~singlenode, [144](#)
 - analyzeNonOptLoops, [144](#)
 - analyzeOptLoops, [144](#)
 - body, [148](#)
 - clone, [144](#)
 - drawToLeftRail, [145](#)
 - drawToRightRail, [145](#)
 - fixSkips, [145](#)
 - forgetChild, [145](#)
 - getChild, [145](#)
 - liftConcats, [146](#)
 - mergeChoices, [146](#)
 - mergeConcats, [146](#)
 - mergeRails, [146](#)
 - numChildren, [146](#)
 - operator!=, [146](#)
 - operator==, [147](#)
 - setNext, [147](#)
 - setParent, [147](#)
 - setPrevious, [147](#)
 - singlenode, [144](#)
 - subsume, [147](#)
 - texName, [148](#)
- SIZE_MAX
 - annot_lexer.cc, [184](#)
 - annot_lexer.hh, [212](#)
 - lexer.cc, [250](#)
 - lexer.hh, [273](#)
- sizes
 - node, [95](#)
- slice
 - annot::parser::stack< T, S >::slice, [149](#)
 - yy::parser::stack< T, S >::slice, [150](#)
- step
 - annot::location, [61](#)
 - annot_lexer.cc, [201](#)
 - lexer.cc, [261](#)
 - yy::location, [65](#)
- str
 - nontermnode, [102](#)
- stripquotes
 - annot_lexer.cc, [201](#)
- stripSpecial
 - graph.cc, [241](#)
 - graph.hh, [245](#)
- style
 - nontermnode, [102](#)
- subloc
 - lexer.cc, [267](#)
- subsume
 - grammar, [58](#)
 - multinode, [75](#)
 - node, [92](#)
 - nontermnode, [102](#)
 - productionnode, [129](#)
 - singlenode, [147](#)
- subsume.cc, [299](#)
 - ARRAY_SIZE, [300](#)
- super_type
 - annot::parser::basic_symbol< Base >, [27](#)
 - annot::parser::symbol_type, [154](#)
 - yy::parser::basic_symbol< Base >, [32](#)
 - yy::parser::symbol_type, [156](#)
- swap
 - annot::parser::semantic_type, [141](#)
 - yy::parser::value_type, [170](#)
- symbol_kind_type
 - annot::parser, [107](#)
 - annot::parser::symbol_kind, [151](#)
 - yy::parser, [114](#)
 - yy::parser::symbol_kind, [152](#)
- symbol_name
 - annot::parser, [111](#)
 - yy::parser, [120](#)
- symbol_type
 - annot::parser::symbol_type, [154](#), [155](#)
 - yy::parser::symbol_type, [157](#)
- syntax_error
 - annot::parser::syntax_error, [159](#)
 - yy::parser::syntax_error, [161](#)
- TERMINAL
 - node, [83](#)
- termnode, [162](#)
 - ~termnode, [163](#)
 - clone, [163](#)
 - termnode, [163](#)
- texName
 - choicenode, [45](#)
 - loopnode, [69](#)
 - multinode, [76](#)
 - node, [92](#)
 - nontermnode, [102](#)
 - nullnode, [105](#)
 - productionnode, [129](#)
 - railnode, [134](#)
 - rownode, [137](#)
 - singlenode, [148](#)

- TOK_AEND
 - annot::parser::token, [164](#)
- TOK_ANNOTATION
 - yy::parser::token, [166](#)
- TOK_ANNOTEMPTY
 - annot::parser::token, [164](#)
- TOK_ANNOTerror
 - annot::parser::token, [164](#)
- TOK_ANNOTUNDEF
 - annot::parser::token, [164](#)
- TOK_AS
 - annot::parser::token, [164](#)
- TOK_ASTART
 - annot::parser::token, [164](#)
- TOK_CAPTION
 - annot::parser::token, [164](#)
- TOK_COMMA
 - yy::parser::token, [166](#)
- TOK_END
 - annot::parser::token, [164](#)
 - yy::parser::token, [166](#)
- TOK_EQUAL
 - yy::parser::token, [166](#)
- TOK_LBRACE
 - yy::parser::token, [166](#)
- TOK_LBRACK
 - yy::parser::token, [166](#)
- TOK_LPAREN
 - yy::parser::token, [166](#)
- TOK_NEWLINE
 - yy::parser::token, [166](#)
- TOK_PIPE
 - yy::parser::token, [166](#)
- TOK_RBRACE
 - yy::parser::token, [166](#)
- TOK_RBRACK
 - yy::parser::token, [166](#)
- TOK_RPAREN
 - yy::parser::token, [166](#)
- TOK_SEMICOLON
 - annot::parser::token, [164](#)
 - yy::parser::token, [166](#)
- TOK_SIDEWAYS
 - annot::parser::token, [164](#)
- TOK_STRING
 - annot::parser::token, [164](#)
 - yy::parser::token, [166](#)
- TOK_SUBSUME
 - annot::parser::token, [164](#)
- TOK_TERM
 - yy::parser::token, [166](#)
- TOK_UNEXP
 - annot::parser::token, [164](#)
 - yy::parser::token, [166](#)
- TOK_YYEMPTY
 - yy::parser::token, [166](#)
- TOK_YYerror
 - yy::parser::token, [166](#)
- TOK_YYUNDEF
 - yy::parser::token, [166](#)
- token
 - annot::parser::context, [51](#)
 - yy::parser::context, [52](#)
- token_kind_type
 - annot::parser, [107](#)
 - annot::parser::token, [164](#)
 - yy::parser, [114](#)
 - yy::parser::token, [165](#)
- token_type
 - annot::parser, [107](#)
 - yy::parser, [115](#)
- top
 - railnode, [134](#)
- type
 - node, [95](#)
- type_get
 - annot::parser::basic_symbol< Base >, [29](#)
 - annot::parser::by_kind, [38](#)
 - yy::parser::basic_symbol< Base >, [35](#)
 - yy::parser::by_kind, [40](#)
- UINT16_MAX
 - annot_lexer.cc, [184](#)
 - annot_lexer.hh, [212](#)
 - lexer.cc, [251](#)
 - lexer.hh, [273](#)
- UINT32_MAX
 - annot_lexer.cc, [185](#)
 - annot_lexer.hh, [212](#)
 - lexer.cc, [251](#)
 - lexer.hh, [273](#)
- UINT8_MAX
 - annot_lexer.cc, [185](#)
 - annot_lexer.hh, [212](#)
 - lexer.cc, [251](#)
 - lexer.hh, [273](#)
- UNKNOWN
 - node, [83](#)
- unput
 - annot_lexer.cc, [185](#)
 - lexer.cc, [251](#)
- UP
 - graph.hh, [244](#)
- usage
 - main.cc, [286](#)
- util.cc, [300](#)
 - camelcase, [300](#)
- util.hh, [301](#)
 - camelcase, [301](#)
- value
 - annot::parser::basic_symbol< Base >, [30](#)
 - yy::parser::basic_symbol< Base >, [35](#)
- value_type
 - yy::parser::value_type, [167](#), [168](#)
- vraildir
 - graph.hh, [243](#)

- vrailside
 - graph.hh, [244](#)
- vrailStr
 - node, [92](#)
- wa
 - node, [95](#)
- west
 - node, [93](#)
- while
 - annot_lexer.cc, [201](#)
 - lexer.cc, [261](#)
- width
 - node, [93](#)
- wrapChoice
 - parser.cc, [293](#)
- x
 - coordinate, [54](#)
- y
 - coordinate, [54](#)
- yy, [20](#)
 - operator<<, [23](#)
 - operator+, [21](#)
 - operator+==, [21](#), [22](#)
 - operator-, [22](#)
 - operator==, [22](#), [23](#)
- yy::location, [62](#)
 - begin, [65](#)
 - columns, [64](#)
 - counter_type, [63](#)
 - end, [65](#)
 - filename_type, [63](#)
 - initialize, [64](#)
 - lines, [65](#)
 - location, [64](#)
 - step, [65](#)
- yy::parser, [112](#)
 - ~parser, [115](#)
 - by_type, [114](#)
 - debug_level, [115](#)
 - debug_level_type, [114](#)
 - debug_stream, [115](#)
 - error, [115](#), [116](#)
 - location_type, [114](#)
 - make_ANNOTATION, [116](#)
 - make_COMMA, [116](#)
 - make_END, [116](#)
 - make_EQUAL, [116](#)
 - make_LBRACE, [116](#)
 - make_LBRACK, [117](#)
 - make_LPAREN, [117](#)
 - make_NEWLINE, [117](#)
 - make_PIPE, [117](#)
 - make_RBRACE, [117](#)
 - make_RBRACK, [117](#)
 - make_LPAREN, [117](#)
 - make_SEMICOLON, [118](#)
 - make_STRING, [118](#)
 - make_TERM, [118](#)
 - make_UNEXP, [118](#)
 - make_YYerror, [118](#)
 - make_YYUNDEF, [118](#)
 - operator(), [119](#)
 - parse, [119](#)
 - parser, [115](#)
 - semantic_type, [114](#)
 - set_debug_level, [119](#)
 - set_debug_stream, [119](#)
 - symbol_kind_type, [114](#)
 - symbol_name, [120](#)
 - token_kind_type, [114](#)
 - token_type, [115](#)
 - YYNTOKENS, [120](#)
- yy::parser::basic_symbol< Base >, [30](#)
 - ~basic_symbol, [34](#)
 - basic_symbol, [32–34](#)
 - clear, [34](#)
 - empty, [34](#)
 - location, [35](#)
 - move, [34](#)
 - name, [35](#)
 - super_type, [32](#)
 - type_get, [35](#)
 - value, [35](#)
- yy::parser::by_kind, [38](#)
 - by_kind, [39](#), [40](#)
 - clear, [40](#)
 - kind, [40](#)
 - kind_, [41](#)
 - kind_type, [39](#)
 - move, [40](#)
 - type_get, [40](#)
- yy::parser::context, [51](#)
 - context, [51](#)
 - expected_tokens, [52](#)
 - location, [52](#)
 - lookahead, [52](#)
 - token, [52](#)
- yy::parser::stack< T, S >::slice, [149](#)
 - operator[], [150](#)
 - slice, [150](#)
- yy::parser::symbol_kind, [151](#)
 - S_ANNOTATION, [152](#)
 - S_annotations, [152](#)
 - S_COMMA, [152](#)
 - S_EQUAL, [152](#)
 - S_expression, [152](#)
 - S_grammar, [152](#)
 - S_LBRACE, [152](#)
 - S_LBRACK, [152](#)
 - S_LPAREN, [152](#)
 - S_NEWLINE, [152](#)
 - S_PIPE, [152](#)
 - S_primary, [152](#)
 - S_production, [152](#)

- S_productions, 152
- S_RBRACE, 152
- S_RBRACK, 152
- S_rows, 152
- S_RPAREN, 152
- S_SEMICOLON, 152
- S_STRING, 152
- S_TERM, 152
- S_UNEXP, 152
- S_YYACCEPT, 152
- S_YYEMPTY, 152
- S_YYEOF, 152
- S_YYerror, 152
- S_YYUNDEF, 152
- symbol_kind_type, 152
- YYTOKENS, 152
- yy::parser::symbol_type, 155
 - super_type, 156
 - symbol_type, 157
- yy::parser::syntax_error, 160
 - ~syntax_error, 161
 - location, 161
 - syntax_error, 161
- yy::parser::token, 165
 - TOK_ANNOTATION, 166
 - TOK_COMMA, 166
 - TOK_END, 166
 - TOK_EQUAL, 166
 - TOK_LBRACE, 166
 - TOK_LBRACK, 166
 - TOK_LPAREN, 166
 - TOK_NEWLINE, 166
 - TOK_PIPE, 166
 - TOK_RBRACE, 166
 - TOK_RBRACK, 166
 - TOK_RPAREN, 166
 - TOK_SEMICOLON, 166
 - TOK_STRING, 166
 - TOK_TERM, 166
 - TOK_UNEXP, 166
 - TOK_YYEMPTY, 166
 - TOK_YYerror, 166
 - TOK_YYUNDEF, 166
 - token_kind_type, 165
 - yytokentype, 165
- yy::parser::value_type, 166
 - ~value_type, 168
 - as, 168
 - build, 168, 169
 - copy, 169
 - destroy, 169
 - emplace, 169
 - move, 169
 - self_type, 167
 - swap, 170
 - value_type, 167, 168
 - yyalign_me_, 170
 - yyraw_, 170
- yy::position, 123
 - column, 125
 - columns, 125
 - counter_type, 124
 - filename, 125
 - filename_type, 124
 - initialize, 125
 - line, 125
 - lines, 125
 - position, 124
- YY_
 - annot_parser.cc, 230
 - parser.cc, 291
- yy_act
 - annot_lexer.cc, 206
 - lexer.cc, 267
- YY_ASSERT
 - parser.hh, 296
- YY_AT_BOL
 - annot_lexer.cc, 185
 - lexer.cc, 251
- yy_at_bol
 - yy_buffer_state, 171
- YY_ATTRIBUTE_PURE
 - annot_parser.hh, 235
 - parser.hh, 296
- YY_ATTRIBUTE_UNUSED
 - annot_parser.hh, 235
 - parser.hh, 296
- yy_bp
 - annot_lexer.cc, 206
 - lexer.cc, 267
- YY_BREAK
 - annot_lexer.cc, 185
 - lexer.cc, 251
- yy_bs_column
 - yy_buffer_state, 171
- yy_bs_lineno
 - yy_buffer_state, 171
- yy_buf_pos
 - yy_buffer_state, 171
- YY_BUF_SIZE
 - annot_lexer.cc, 185
 - annot_lexer.hh, 212
 - lexer.cc, 251
 - lexer.hh, 273
- yy_buf_size
 - yy_buffer_state, 171
- YY_BUFFER_EOF_PENDING
 - annot_lexer.cc, 185
 - lexer.cc, 252
- YY_BUFFER_NEW
 - annot_lexer.cc, 186
 - lexer.cc, 252
- YY_BUFFER_NORMAL
 - annot_lexer.cc, 186
 - lexer.cc, 252
- YY_BUFFER_STATE

- annot_lexer.cc, [200](#)
- annot_lexer.hh, [221](#)
- lexer.cc, [260](#)
- lexer.hh, [276](#)
- yy_buffer_state, [170](#)
- yy_at_bol, [171](#)
- yy_bs_column, [171](#)
- yy_bs_lineno, [171](#)
- yy_buf_pos, [171](#)
- yy_buf_size, [171](#)
- yy_buffer_status, [171](#)
- yy_ch_buf, [172](#)
- yy_fill_buffer, [172](#)
- yy_input_file, [172](#)
- yy_is_interactive, [172](#)
- yy_is_our_buffer, [172](#)
- yy_n_chars, [172](#)
- yy_buffer_status
 - yy_buffer_state, [171](#)
- YY_CAST
 - annot_parser.hh, [235](#)
 - parser.hh, [296](#)
- yy_ch_buf
 - yy_buffer_state, [172](#)
- YY_CHAR
 - annot_lexer.cc, [200](#)
 - lexer.cc, [260](#)
- YY_CONSTEXPR
 - annot_parser.hh, [236](#)
 - parser.hh, [296](#)
- YY_COPY
 - annot_parser.hh, [236](#)
 - parser.hh, [296](#)
- yy_cp
 - annot_lexer.cc, [206](#)
 - lexer.cc, [267](#)
- YY_CPLUSPLUS
 - annot_parser.hh, [236](#)
 - parser.hh, [296](#)
- yy_create_buffer
 - annot_lexer.cc, [186](#), [201](#)
 - annot_lexer.hh, [212](#), [221](#)
 - lexer.cc, [261](#)
 - lexer.hh, [277](#)
- YY_CURRENT_BUFFER
 - annot_lexer.cc, [186](#)
 - lexer.cc, [252](#)
- YY_CURRENT_BUFFER_LVALUE
 - annot_lexer.cc, [186](#)
 - lexer.cc, [252](#)
- YY_DECL
 - annot_lexer.cc, [186](#), [206](#)
 - annot_lexer.hh, [213](#)
 - annot_parser.cc, [230](#), [233](#)
 - driver.hh, [240](#)
 - lexer.cc, [268](#)
 - lexer.hh, [273](#)
- YY_DECL_IS_OURS
 - annot_lexer.hh, [213](#)
 - lexer.hh, [273](#)
- lexer.hh, [273](#)
- yy_delete_buffer
 - annot_lexer.cc, [186](#), [202](#)
 - annot_lexer.hh, [213](#), [221](#)
 - lexer.cc, [261](#)
 - lexer.hh, [277](#)
- YY_DO_BEFORE_ACTION
 - annot_lexer.cc, [187](#)
 - lexer.cc, [252](#)
- YY_END_OF_BUFFER
 - annot_lexer.cc, [187](#)
 - lexer.cc, [252](#)
- YY_END_OF_BUFFER_CHAR
 - annot_lexer.cc, [187](#)
 - lexer.cc, [253](#)
- YY_EXCEPTIONS
 - annot_parser.cc, [230](#)
 - parser.cc, [291](#)
- YY_EXIT_FAILURE
 - annot_lexer.cc, [187](#)
 - lexer.cc, [253](#)
- YY_EXTRA_TYPE
 - annot_lexer.cc, [187](#)
 - annot_lexer.hh, [213](#)
 - lexer.cc, [253](#)
 - lexer.hh, [273](#)
- YY_FATAL_ERROR
 - annot_lexer.cc, [187](#)
 - lexer.cc, [253](#)
- yy_fill_buffer
 - yy_buffer_state, [172](#)
- yy_flex_debug
 - annot_lexer.cc, [187](#), [206](#)
 - annot_lexer.hh, [213](#)
 - lexer.cc, [268](#)
- YY_FLEX_MAJOR_VERSION
 - annot_lexer.cc, [188](#)
 - annot_lexer.hh, [213](#)
 - lexer.cc, [253](#)
 - lexer.hh, [274](#)
- YY_FLEX_MINOR_VERSION
 - annot_lexer.cc, [188](#)
 - annot_lexer.hh, [213](#)
 - lexer.cc, [253](#)
 - lexer.hh, [274](#)
- YY_FLEX_SUBMINOR_VERSION
 - annot_lexer.cc, [188](#)
 - annot_lexer.hh, [213](#)
 - lexer.cc, [253](#)
 - lexer.hh, [274](#)
- YY_FLUSH_BUFFER
 - annot_lexer.cc, [188](#)
 - lexer.cc, [254](#)
- yy_flush_buffer
 - annot_lexer.cc, [188](#), [202](#)
 - annot_lexer.hh, [214](#), [222](#)
 - lexer.cc, [262](#)

- lexer.hh, 277
- YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN
 - annot_parser.hh, 236
 - parser.hh, 297
- YY_IGNORE_MAYBE_UNINITIALIZED_END
 - annot_parser.hh, 236
 - parser.hh, 297
- YY_IGNORE_USELESS_CAST_BEGIN
 - annot_parser.hh, 236
 - parser.hh, 297
- YY_IGNORE_USELESS_CAST_END
 - annot_parser.hh, 236
 - parser.hh, 297
- yy_init_buffer
 - annot_lexer.cc, 188
 - annot_lexer.hh, 214
- YY_INITIAL_VALUE
 - annot_parser.hh, 237
 - parser.hh, 297
- YY_INPUT
 - annot_lexer.cc, 188
 - lexer.cc, 254
- yy_input_file
 - yy_buffer_state, 172
- YY_INT_ALIGNED
 - annot_lexer.cc, 189
 - annot_lexer.hh, 214
 - lexer.cc, 254
 - lexer.hh, 274
- yy_is_interactive
 - yy_buffer_state, 172
- yy_is_our_buffer
 - yy_buffer_state, 172
- YY_LESS_LINENO
 - annot_lexer.cc, 189
 - lexer.cc, 254
- YY_LINENO_REWIND_TO
 - annot_lexer.cc, 189
 - lexer.cc, 254
- yy_load_buffer_state
 - annot_lexer.cc, 189
 - annot_lexer.hh, 214
- YY_MORE_ADJ
 - annot_lexer.cc, 190
 - lexer.cc, 254
- YY_MOVE
 - annot_parser.hh, 237
 - parser.hh, 297
- YY_MOVE_OR_COPY
 - annot_parser.hh, 237
 - parser.hh, 297
- YY_MOVE_REF
 - annot_parser.hh, 237
 - parser.hh, 298
- yy_n_chars
 - yy_buffer_state, 172
- yy_new_buffer
 - annot_lexer.cc, 190
- lexer.cc, 254
- YY_NEW_FILE
 - annot_lexer.cc, 190
 - lexer.cc, 255
- YY_NO_INPUT
 - annot_lexer.cc, 190
 - lexer.cc, 255
- YY_NOEXCEPT
 - annot_parser.hh, 237
 - parser.hh, 298
- YY_NOTHROW
 - annot_parser.hh, 237
 - parser.hh, 298
- YY_NULL
 - annot_lexer.cc, 190
 - lexer.cc, 255
- YY_NULLPTR
 - annot_location.hh, 228
 - location.hh, 285
- YY_NUM_RULES
 - annot_lexer.cc, 190
 - lexer.cc, 255
- yy_nxt
 - yy_trans_info, 173
- YY_READ_BUF_SIZE
 - annot_lexer.cc, 190
 - annot_lexer.hh, 214
 - lexer.cc, 255
 - lexer.hh, 274
- YY_REDUCE_PRINT
 - annot_parser.cc, 230
 - parser.cc, 291
- YY_REINTERPRET_CAST
 - annot_parser.hh, 237
 - parser.hh, 298
- YY_RESTORE_YY_MORE_OFFSET
 - annot_lexer.cc, 191
 - lexer.cc, 255
- YY_RULE_SETUP
 - annot_lexer.cc, 191
 - lexer.cc, 255
- YY_RVREF
 - annot_parser.hh, 238
 - parser.hh, 298
- YY_SC_TO_UI
 - annot_lexer.cc, 191
 - lexer.cc, 255
- yy_scan_buffer
 - annot_lexer.cc, 191, 202
 - annot_lexer.hh, 214, 222
 - lexer.cc, 262
 - lexer.hh, 278
- yy_scan_bytes
 - annot_lexer.cc, 191, 203
 - annot_lexer.hh, 214, 222
 - lexer.cc, 262
 - lexer.hh, 278
- yy_scan_string

- annot_lexer.cc, [191](#), [203](#)
- annot_lexer.hh, [215](#), [223](#)
- lexer.cc, [263](#)
- lexer.hh, [278](#)
- yy_set_bol
 - annot_lexer.cc, [191](#)
 - lexer.cc, [256](#)
- yy_set_interactive
 - annot_lexer.cc, [192](#)
 - lexer.cc, [256](#)
- yy_size_t
 - annot_lexer.cc, [200](#)
 - annot_lexer.hh, [221](#)
 - lexer.cc, [260](#)
 - lexer.hh, [276](#)
- YY_SKIP_YYWRAP
 - annot_lexer.cc, [192](#)
 - annot_lexer.hh, [215](#)
 - lexer.cc, [256](#)
 - lexer.hh, [274](#)
- YY_STACK_PRINT
 - annot_parser.cc, [230](#)
 - parser.cc, [291](#)
- YY_START
 - annot_lexer.cc, [192](#)
 - lexer.cc, [256](#)
- YY_START_STACK_INCR
 - annot_lexer.cc, [192](#)
 - annot_lexer.hh, [215](#)
 - lexer.cc, [256](#)
 - lexer.hh, [274](#)
- YY_STATE_BUF_SIZE
 - annot_lexer.cc, [192](#)
 - lexer.cc, [257](#)
- YY_STATE_EOF
 - annot_lexer.cc, [193](#)
 - lexer.cc, [257](#)
- yy_state_type
 - annot_lexer.cc, [200](#)
 - lexer.cc, [260](#)
- YY_STRUCT_YY_BUFFER_STATE
 - annot_lexer.cc, [193](#)
 - annot_lexer.hh, [215](#)
 - lexer.cc, [257](#)
 - lexer.hh, [274](#)
- yy_switch_to_buffer
 - annot_lexer.cc, [193](#), [203](#)
 - annot_lexer.hh, [215](#), [223](#)
 - lexer.cc, [263](#)
 - lexer.hh, [279](#)
- YY_SYMBOL_PRINT
 - annot_parser.cc, [230](#)
 - parser.cc, [291](#)
- yy_trans_info, [173](#)
 - yy_nxt, [173](#)
 - yy_verify, [173](#)
- YY_TYPEDEF_YY_BUFFER_STATE
 - annot_lexer.cc, [193](#)
 - annot_lexer.hh, [215](#)
 - lexer.cc, [263](#)
 - lexer.hh, [278](#)
- YY_TYPEDEF_YY_SIZE_T
 - annot_lexer.cc, [193](#)
 - annot_lexer.hh, [215](#)
 - lexer.cc, [257](#)
 - lexer.hh, [275](#)
- YY_USE
 - annot_parser.hh, [238](#)
 - parser.hh, [298](#)
- YY_USER_ACTION
 - annot_lexer.cc, [193](#)
 - lexer.cc, [257](#)
- yy_verify
 - yy_trans_info, [173](#)
- YYABORT
 - annot_parser.cc, [230](#)
 - parser.cc, [291](#)
- YYACCEPT
 - annot_parser.cc, [231](#)
 - parser.cc, [292](#)
- yyalign_me
 - annot::parser::semantic_type, [142](#)
- yyalign_me_
 - yy::parser::value_type, [170](#)
- yyalloc
 - annot_lexer.cc, [193](#), [204](#)
 - annot_lexer.hh, [215](#), [223](#)
 - lexer.cc, [263](#)
 - lexer.hh, [279](#)
- YYCASE_
 - annot_parser.cc, [231](#)
 - parser.cc, [292](#)
- YYCDEBUG
 - annot_parser.cc, [231](#)
 - parser.cc, [292](#)
- yyclearin
 - annot_parser.cc, [231](#)
 - parser.cc, [292](#)
- yyconst
 - annot_lexer.cc, [194](#)
 - annot_lexer.hh, [216](#)
 - lexer.cc, [257](#)
 - lexer.hh, [275](#)
- YYDEBUG
 - parser.hh, [298](#)
- yyensure_buffer_stack
 - annot_lexer.cc, [194](#)
 - annot_lexer.hh, [216](#)
- yyerror
 - annot_parser.cc, [231](#)
 - parser.cc, [292](#)
- YYERROR
 - annot_parser.cc, [231](#)
 - parser.cc, [292](#)
- yyfree
 - annot_lexer.cc, [194](#), [204](#)

- annot_lexer.hh, [216](#), [224](#)
 - lexer.cc, [264](#)
 - lexer.hh, [279](#)
- yyget_debug
 - annot_lexer.cc, [194](#)
 - annot_lexer.hh, [216](#)
 - lexer.cc, [264](#)
 - lexer.hh, [279](#)
- yyget_extra
 - annot_lexer.cc, [194](#)
 - annot_lexer.hh, [216](#)
 - lexer.cc, [264](#)
 - lexer.hh, [279](#)
- yyget_in
 - annot_lexer.cc, [194](#)
 - annot_lexer.hh, [216](#)
 - lexer.cc, [264](#)
 - lexer.hh, [280](#)
- yyget_leng
 - annot_lexer.cc, [194](#)
 - annot_lexer.hh, [216](#)
 - lexer.cc, [264](#)
 - lexer.hh, [280](#)
- yyget_lineno
 - annot_lexer.cc, [195](#)
 - annot_lexer.hh, [217](#)
 - lexer.cc, [264](#)
 - lexer.hh, [280](#)
- yyget_out
 - annot_lexer.cc, [195](#)
 - annot_lexer.hh, [217](#)
 - lexer.cc, [264](#)
 - lexer.hh, [280](#)
- yyget_text
 - annot_lexer.cc, [195](#)
 - annot_lexer.hh, [217](#)
 - lexer.cc, [265](#)
 - lexer.hh, [280](#)
- yyin
 - annot_lexer.cc, [195](#), [207](#)
 - annot_lexer.hh, [217](#), [226](#)
 - lexer.cc, [268](#)
 - lexer.hh, [283](#)
- yyIN_HEADER
 - lexer.hh, [275](#)
- yyleng
 - annot_lexer.cc, [195](#), [207](#)
 - annot_lexer.hh, [217](#), [226](#)
 - lexer.cc, [268](#)
 - lexer.hh, [283](#)
- yyless
 - annot_lexer.cc, [195](#), [196](#)
 - lexer.cc, [258](#)
- yylex
 - annot_lexer.cc, [196](#)
 - annot_lexer.hh, [217](#)
 - annot_parser.cc, [231](#)
 - lexer.hh, [280](#)
- yylex_destroy
 - annot_lexer.cc, [196](#)
 - annot_lexer.hh, [217](#)
 - lexer.cc, [265](#)
 - lexer.hh, [280](#)
- yylex_init
 - annot_lexer.cc, [196](#)
 - annot_lexer.hh, [218](#)
- yylex_init_extra
 - annot_lexer.cc, [196](#)
 - annot_lexer.hh, [218](#)
- yylineno
 - annot_lexer.cc, [197](#), [207](#)
 - annot_lexer.hh, [218](#), [226](#)
 - lexer.cc, [268](#)
 - lexer.hh, [283](#)
- YYLOC_DEFAULT
 - annot_parser.cc, [232](#)
 - parser.cc, [292](#)
- yymore
 - annot_lexer.cc, [197](#)
 - lexer.cc, [258](#)
- yynoreturn
 - annot_lexer.cc, [197](#)
 - annot_lexer.hh, [218](#)
 - lexer.cc, [258](#)
 - lexer.hh, [275](#)
- YYNTOKENS
 - annot::parser, [111](#)
 - annot::parser::symbol_kind, [151](#)
 - yy::parser, [120](#)
 - yy::parser::symbol_kind, [152](#)
- yyout
 - annot_lexer.cc, [197](#), [207](#)
 - annot_lexer.hh, [218](#), [226](#)
 - lexer.cc, [268](#)
 - lexer.hh, [283](#)
- yypop_buffer_state
 - annot_lexer.cc, [197](#)
 - annot_lexer.hh, [218](#)
 - lexer.cc, [265](#)
 - lexer.hh, [281](#)
- yypush_buffer_state
 - annot_lexer.cc, [197](#), [204](#)
 - annot_lexer.hh, [218](#), [224](#)
 - lexer.cc, [265](#)
 - lexer.hh, [281](#)
- yyraw
 - annot::parser::semantic_type, [142](#)
- yyraw_
 - yy::parser::value_type, [170](#)
- yyrealloc
 - annot_lexer.cc, [197](#), [204](#)
 - annot_lexer.hh, [219](#), [224](#)
 - lexer.cc, [265](#)
 - lexer.hh, [281](#)
- YYRECOVERING
 - annot_parser.cc, [232](#)

- parser.cc, [293](#)
- yyrestart
 - annot_lexer.cc, [198](#), [204](#)
 - annot_lexer.hh, [219](#), [224](#)
 - lexer.cc, [265](#)
 - lexer.hh, [281](#)
- YYRHSLOC
 - annot_parser.cc, [232](#)
 - parser.cc, [293](#)
- yyset_debug
 - annot_lexer.cc, [198](#), [205](#)
 - annot_lexer.hh, [219](#), [225](#)
 - lexer.cc, [266](#)
 - lexer.hh, [282](#)
- yyset_extra
 - annot_lexer.cc, [198](#), [205](#)
 - annot_lexer.hh, [219](#), [225](#)
 - lexer.cc, [266](#)
 - lexer.hh, [282](#)
- yyset_in
 - annot_lexer.cc, [198](#), [205](#)
 - annot_lexer.hh, [219](#), [225](#)
 - lexer.cc, [266](#)
 - lexer.hh, [282](#)
- yyset_lineno
 - annot_lexer.cc, [198](#), [205](#)
 - annot_lexer.hh, [219](#), [225](#)
 - lexer.cc, [266](#)
 - lexer.hh, [282](#)
- yyset_out
 - annot_lexer.cc, [198](#), [206](#)
 - annot_lexer.hh, [219](#), [226](#)
 - lexer.cc, [267](#)
 - lexer.hh, [283](#)
- YYSTATE
 - annot_lexer.cc, [198](#)
 - lexer.cc, [258](#)
- YYTABLES_NAME
 - annot_lexer.cc, [198](#)
 - lexer.cc, [259](#)
- yyterminate
 - annot_lexer.cc, [199](#)
 - lexer.cc, [259](#)
- yytext
 - annot_lexer.cc, [199](#), [207](#)
 - annot_lexer.hh, [219](#), [226](#)
 - lexer.cc, [268](#)
 - lexer.hh, [283](#)
- yytext_ptr
 - annot_lexer.cc, [199](#)
 - annot_lexer.hh, [220](#)
 - lexer.cc, [259](#)
 - lexer.hh, [275](#)
- yytokentype
 - annot::parser::token, [164](#)
 - yy::parser::token, [165](#)
- yywrap
 - annot_lexer.cc, [199](#)
- annot_lexer.hh, [220](#)
- lexer.cc, [259](#)
- lexer.hh, [275](#)