Klocwork C/C++ Abstract Syntax Tree API

1.9.0

Generated by Doxygen 1.8.18

1 Deprecated List 1

2 Module Index 3

2.1 Modules . . . 3

3 File Index 5

3.1 File List . . . 5

4 Module Documentation 7

4.1 Obtaining configuration parameters for an error . . . 7

4.2 Basic Abstract Syntax Tree traversal and checking routines . . . 8

4.3 Accessing node stack . . . 9

4.4 Setting and clearing handlers for events during tree traversal . . . 10

4.5 Access to semantic information . . . 11

4.6 Functions for accessing type information . . . 12

4.7 Attribute values for signedness of types . . . 13

4.8 Utility functions for tree access . . . 14

4.9 Working with tree positions . . . 15

4.10 for defect . . . 16

4.11 Accessing compiler configuration . . . 17

4.12 Working with warning and error messages . . . 18

4.13 Tree type identifiers . . . 19

4.14 Tree type checking predicates . . . 20

4.15 Child link identifiers . . . 21

4.16 Numerical codes of operations . . . 22

4.16.1 Detailed Description . . . 23

4.16.2 Variable Documentation . . . 23

4.16.2.1 KTC_OPCODE_ADD . . . 23

4.16.2.2 KTC_OPCODE_ADDASSIGN . . . 23

4.16.2.3 KTC_OPCODE_ADDRESS . . . 23

4.16.2.4 KTC_OPCODE_ANDASSIGN . . . 23

4.16.2.5 KTC_OPCODE_ASL . . . 23

4.16.2.6 KTC_OPCODE_ASLASSIGN . . . 24

4.16.2.7 KTC_OPCODE_ASR . . . 24

4.16.2.8 KTC_OPCODE_ASRASSIGN . . . 24

4.16.2.9 KTC_OPCODE_ASSIGN . . . 24

4.16.2.10 KTC_OPCODE_BITAND . . . 24

4.16.2.11 KTC_OPCODE_BITNOT . . . 24

4.16.2.12 KTC_OPCODE_BITOR . . . 24

4.16.2.13 KTC_OPCODE_BITXOR . . . 24

4.16.2.14 KTC_OPCODE_COMMA . . . 25

4.16.2.15 KTC_OPCODE_COND . . . 25

4.16.2.16 KTC_OPCODE_DEREF . . . 25

4.16.2.17 KTC_OPCODE_DEREFAST . . . 25
4.16.2.18 KTC_OPCODE_DIV . . . 25
4.16.2.19 KTC_OPCODE_DIVASSIGN . . . 25
4.16.2.20 KTC_OPCODE_DOTAST . . . 25
4.16.2.21 KTC_OPCODE_EQ . . . 25
4.16.2.22 KTC_OPCODE_FIELD . . . 26
4.16.2.23 KTC_OPCODE_FIELDREF . . . 26
4.16.2.24 KTC_OPCODE_GE . . . 26
4.16.2.25 KTC_OPCODE_GT . . . 26
4.16.2.26 KTC_OPCODE_LE . . . 26
4.16.2.27 KTC_OPCODE_LOGAND . . . 26
4.16.2.28 KTC_OPCODE_LOGNOT . . . 26
4.16.2.29 KTC_OPCODE_LOGOR . . . 26
4.16.2.30 KTC_OPCODE_LT . . . 27
4.16.2.31 KTC_OPCODE_MAX . . . 27
4.16.2.32 KTC_OPCODE_MIN . . . 27
4.16.2.33 KTC_OPCODE_MINUS . . . 27
4.16.2.34 KTC_OPCODE_MOD . . . 27
4.16.2.35 KTC_OPCODE_MODASSIGN . . . 27
4.16.2.36 KTC_OPCODE_MUL . . . 27
4.16.2.37 KTC_OPCODE_MULASSIGN . . . 27
4.16.2.38 KTC_OPCODE_NE . . . 28
4.16.2.39 KTC_OPCODE_NONE . . . 28
4.16.2.40 KTC_OPCODE_ORASSIGN . . . 28
4.16.2.41 KTC_OPCODE_PLUS . . . 28
4.16.2.42 KTC_OPCODE_POSTDEC . . . 28
4.16.2.43 KTC_OPCODE_POSTINC . . . 28
4.16.2.44 KTC_OPCODE_PREDEC . . . 28
4.16.2.45 KTC_OPCODE_PREINC . . . 28
4.16.2.46 KTC_OPCODE_ROUND_BRACKETS . . . 29
4.16.2.47 KTC_OPCODE_SIZEOF . . . 29
4.16.2.48 KTC_OPCODE_SQUARE_BRACKETS . . . 29
4.16.2.49 KTC_OPCODE_SUB . . . 29
4.16.2.50 KTC_OPCODE_SUBASSIGN . . . 29
4.16.2.51 KTC_OPCODE_THROW . . . 29
4.16.2.52 KTC_OPCODE_XORASSIGN . . . 29
4.17 Numerical codes of declaration storage class specifiers . . . 30
4.17.1 Detailed Description . . . 30
4.17.2 Variable Documentation . . . 30
4.17.2.1 KTC_STORAGECLASS_AUTO . . . 30
4.17.2.2 KTC_STORAGECLASS_EXTERN . . . 30
4.17.2.3 KTC_STORAGECLASS_MUTABLE . . . 30

4.17.2.4 KTC_STORAGECLASS_NONE . . . 30
4.17.2.5 KTC_STORAGECLASS_REGISTER . . . 31
4.17.2.6 KTC_STORAGECLASS_STATIC . . . 31
4.17.2.7 KTC_STORAGECLASS_THREADLOCAL . . . 31
4.17.2.8 KTC_STORAGECLASS_TYPEDEF . . . 31
4.18 Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values . . . 32
4.18.1 Detailed Description . . . 32
4.18.2 Variable Documentation . . . 32
4.18.2.1 KTC_CVQUALIFIER_CONST . . . 32
4.18.2.2 KTC_CVQUALIFIER_NONE . . . 32
4.18.2.3 KTC_CVQUALIFIER_RESTRICT . . . 32
4.18.2.4 KTC_CVQUALIFIER_VOLATILE . . . 32
4.19 Numerical codes to differentiate struct/class/union declarations . . . 33
4.19.1 Detailed Description . . . 33
4.19.2 Variable Documentation . . . 33
4.19.2.1 KTC_CLASSTAG_CLASS . . . 33
4.19.2.2 KTC_CLASSTAG_NONE . . . 33
4.19.2.3 KTC_CLASSTAG_STRUCT . . . 33
4.19.2.4 KTC_CLASSTAG_UNION . . . 33
4.20 Numerical codes of C/C++ built-in types . . . 34
4.20.1 Detailed Description . . . 34
4.20.2 Variable Documentation . . . 34
4.20.2.1 KTC_BUILTINTYPE_BOOL . . . 34
4.20.2.2 KTC_BUILTINTYPE_CHAR . . . 34
4.20.2.3 KTC_BUILTINTYPE_DOUBLE . . . 35
4.20.2.4 KTC_BUILTINTYPE_FLOAT . . . 35
4.20.2.5 KTC_BUILTINTYPE_INT . . . 35
4.20.2.6 KTC_BUILTINTYPE_LONGDOUBLE . . . 35
4.20.2.7 KTC_BUILTINTYPE_LONGINT . . . 35
4.20.2.8 KTC_BUILTINTYPE_LONGLONGINT . . . 35
4.20.2.9 KTC_BUILTINTYPE_NONE . . . 35
4.20.2.10 KTC_BUILTINTYPE_NULLPTR_T . . . 35
4.20.2.11 KTC_BUILTINTYPE_SHORTINT . . . 36
4.20.2.12 KTC_BUILTINTYPE_SIGNEDCHAR . . . 36
4.20.2.13 KTC_BUILTINTYPE_SIGNEDINT . . . 36
4.20.2.14 KTC_BUILTINTYPE_SIGNEDLONGINT . . . 36
4.20.2.15 KTC_BUILTINTYPE_SIGNEDLONGLONGINT . . . 36
4.20.2.16 KTC_BUILTINTYPE_SIGNEDSHORTINT . . . 36
4.20.2.17 KTC_BUILTINTYPE_UNSIGNEDCHAR . . . 36
4.20.2.18 KTC_BUILTINTYPE_UNSIGNEDINT . . . 36
4.20.2.19 KTC_BUILTINTYPE_UNSIGNEDLONGINT . . . 37

4.20.2.20 KTC_BUILTINTYPE_UNSIGNEDLONGLONGINT . . . 37
4.20.2.21 KTC_BUILTINTYPE_UNSIGNEDSHORTINT . . . 37
4.20.2.22 KTC_BUILTINTYPE_VOID . . . 37
4.20.2.23 KTC_BUILTINTYPE_WCHAR_T . . . 37
4.21 Numerical codes for identifying inline/virtual/explicit/friend member function specifiers . . . 38
4.21.1 Detailed Description . . . 38
4.21.2 Variable Documentation . . . 38
4.21.2.1 KTC_FUNCSPECIFIER_EXPLICIT . . . 38
4.21.2.2 KTC_FUNCSPECIFIER_FRIEND . . . 38
4.21.2.3 KTC_FUNCSPECIFIER_INLINE . . . 38
4.21.2.4 KTC_FUNCSPECIFIER_NONE . . . 38
4.21.2.5 KTC_FUNCSPECIFIER_VIRTUAL . . . 38
4.22 Numerical codes for identifying different C++ -style cast expressions . . . 39
4.22.1 Detailed Description . . . 39
4.22.2 Variable Documentation . . . 39
4.22.2.1 KTC_CASTSPECIFIER_CONST . . . 39
4.22.2.2 KTC_CASTSPECIFIER_DYNAMIC . . . 39
4.22.2.3 KTC_CASTSPECIFIER_REINTERPRET . . . 39
4.22.2.4 KTC_CASTSPECIFIER_STATIC . . . 39
4.23 Numerical codes for identifying declarator ptr types . . . 40
4.23.1 Detailed Description . . . 40
4.23.2 Function Documentation . . . 40
4.23.2.1 ktc_get_child_index() . . . 40
4.23.3 Variable Documentation . . . 40
4.23.3.1 KTC_POINTEROPERATOR_NONE . . . 40
4.23.3.2 KTC_POINTEROPERATOR_POINTER . . . 40
4.23.3.3 KTC_POINTEROPERATOR_REFERENCE . . . 40
4.23.3.4 KTC_POINTEROPERATOR_RVALUE . . . 40

5 File Documentation **41**
5.1 gen-ktcAPI.h File Reference . . . 41
5.1.1 Function Documentation . . . 53
5.1.1.1 ktc_is_AccessSpecification() . . . 53
5.1.1.2 ktc_is_AliasDecl() . . . 54
5.1.1.3 ktc_is_AlignAsExpr() . . . 54
5.1.1.4 ktc_is_AlignAsType() . . . 54
5.1.1.5 ktc_is_AlignOfExpr() . . . 55
5.1.1.6 ktc_is_AnyAttribute() . . . 55
5.1.1.7 ktc_is_AnyCapture() . . . 55
5.1.1.8 ktc_is_AnyDecl() . . . 56
5.1.1.9 ktc_is_AnyDeclarator() . . . 56
5.1.1.10 ktc_is_AnyDesignator() . . . 56

5.1.1.11 ktc_is_AnyEnumerator() . . . 57
5.1.1.12 ktc_is_AnyExpr() . . . 57
5.1.1.13 ktc_is_AnyFuncBody() . . . 57
5.1.1.14 ktc_is_AnyInitializer() . . . 59
5.1.1.15 ktc_is_AnyLabel() . . . 59
5.1.1.16 ktc_is_AnyMemberDecl() . . . 59
5.1.1.17 ktc_is_AnyName() . . . 61
5.1.1.18 ktc_is_AnyNameQualifier() . . . 61
5.1.1.19 ktc_is_AnyNames() . . . 61
5.1.1.20 ktc_is_AnyNameSpec() . . . 63
5.1.1.21 ktc_is_AnyNonPtrDeclarator() . . . 63
5.1.1.22 ktc_is_AnyParamName() . . . 63
5.1.1.23 ktc_is_AnyPropertyFunc() . . . 65
5.1.1.24 ktc_is_AnyPseudoDtor() . . . 65
5.1.1.25 ktc_is_AnyStmt() . . . 65
5.1.1.26 ktc_is_AnyTemplateArg() . . . 67
5.1.1.27 ktc_is_AnyTypeName() . . . 67
5.1.1.28 ktc_is_AnyTypeOf() . . . 67
5.1.1.29 ktc_is_AnyTypeParam() . . . 69
5.1.1.30 ktc_is_AnyUsing() . . . 69
5.1.1.31 ktc_is_ArrayDeclarator() . . . 69
5.1.1.32 ktc_is_AsmDef() . . . 71
5.1.1.33 ktc_is_AsmStmt() . . . 71
5.1.1.34 ktc_is_Attribute() . . . 71
5.1.1.35 ktc_is_AttributedDeclarator() . . . 73
5.1.1.36 ktc_is_AttributeDeclSpec() . . . 73
5.1.1.37 ktc_is_Attributes() . . . 73
5.1.1.38 ktc_is_AttributeSpec() . . . 75
5.1.1.39 ktc_is_AttributeSpecs() . . . 75
5.1.1.40 ktc_is_AttributeWithArgs() . . . 75
5.1.1.41 ktc_is_AutoType() . . . 77
5.1.1.42 ktc_is_BaseSpec() . . . 77
5.1.1.43 ktc_is_BaseSpecs() . . . 77
5.1.1.44 ktc_is_BinaryExpr() . . . 79
5.1.1.45 ktc_is_BitFieldDeclarator() . . . 79
5.1.1.46 ktc_is_BoolLiteralExpr() . . . 79
5.1.1.47 ktc_is_BreakStmt() . . . 81
5.1.1.48 ktc_is_BuiltinType() . . . 81
5.1.1.49 ktc_is_CallExpr() . . . 81
5.1.1.50 ktc_is_Capture() . . . 83
5.1.1.51 ktc_is_CaptureDefault() . . . 83
5.1.1.52 ktc_is_CaseLabel() . . . 83

5.1.1.53 ktc_is_CaseRangeLabel() . . . 85
5.1.1.54 ktc_is_CastExpr() . . . 85
5.1.1.55 ktc_is_ClassType() . . . 85
5.1.1.56 ktc_is_CompoundStmt() . . . 87
5.1.1.57 ktc_is_ConditionalExpr() . . . 87
5.1.1.58 ktc_is_ConstExpr() . . . 87
5.1.1.59 ktc_is_ContinueStmt() . . . 89
5.1.1.60 ktc_is_ConvFunc() . . . 89
5.1.1.61 ktc_is_CopyInitializer() . . . 89
5.1.1.62 ktc_is_CtorInitializer() . . . 91
5.1.1.63 ktc_is_CVQualifier() . . . 91
5.1.1.64 ktc_is_Decl() . . . 91
5.1.1.65 ktc_is_DeclEllipsis() . . . 93
5.1.1.66 ktc_is_DeclOrStmt() . . . 93
5.1.1.67 ktc_is_DeclOrStmts() . . . 93
5.1.1.68 ktc_is_DeclSpec() . . . 95
5.1.1.69 ktc_is_DeclSpecs() . . . 95
5.1.1.70 ktc_is_DefaultException() . . . 95
5.1.1.71 ktc_is_DefaultLabel() . . . 97
5.1.1.72 ktc_is_DeleteExpr() . . . 97
5.1.1.73 ktc_is_DenyThrowSpec() . . . 97
5.1.1.74 ktc_is_Designators() . . . 99
5.1.1.75 ktc_is_DirectInitializer() . . . 99
5.1.1.76 ktc_is_DoDeclStmt() . . . 99
5.1.1.77 ktc_is_DoStmt() . . . 101
5.1.1.78 ktc_is_Dtor() . . . 101
5.1.1.79 ktc_is_Enumerator() . . . 101
5.1.1.80 ktc_is_Enumerators() . . . 103
5.1.1.81 ktc_is_EnumType() . . . 103
5.1.1.82 ktc_is_ExceptHandler() . . . 103
5.1.1.83 ktc_is_Exception() . . . 105
5.1.1.84 ktc_is_ExceptionSpec() . . . 105
5.1.1.85 ktc_is_ExplicitInstantiation() . . . 105
5.1.1.86 ktc_is_ExprArg() . . . 107
5.1.1.87 ktc_is_Exprs() . . . 107
5.1.1.88 ktc_is_ExprStmt() . . . 107
5.1.1.89 ktc_is_ExprTypeIdExpr() . . . 109
5.1.1.90 ktc_is_FieldDesignator() . . . 109
5.1.1.91 ktc_is_FinallyHandler() . . . 109
5.1.1.92 ktc_is_ForEachStmt() . . . 111
5.1.1.93 ktc_is_ForRangeStmt() . . . 111
5.1.1.94 ktc_is_ForStmt() . . . 111

5.1.1.95 ktc_is_FuncBody() . . . 113
5.1.1.96 ktc_is_FuncDeclarator() . . . 113
5.1.1.97 ktc_is_FuncDef() . . . 113
5.1.1.98 ktc_is_FuncSpec() . . . 115
5.1.1.99 ktc_is_FuncTryBlock() . . . 115
5.1.1.100 ktc_is_GenericAttribute() . . . 115
5.1.1.101 ktc_is_GlobalScope() . . . 117
5.1.1.102 ktc_is_GotoStmt() . . . 117
5.1.1.103 ktc_is_Handler() . . . 117
5.1.1.104 ktc_is_Handlers() . . . 119
5.1.1.105 ktc_is_IdExpr() . . . 119
5.1.1.106 ktc_is_IfDeclStmt() . . . 119
5.1.1.107 ktc_is_IfStmt() . . . 121
5.1.1.108 ktc_is_IndexDesignator() . . . 121
5.1.1.109 ktc_is_IndexExpr() . . . 121
5.1.1.110 ktc_is_InitClause() . . . 123
5.1.1.111 ktc_is_InitializedDeclarator() . . . 123
5.1.1.112 ktc_is_InitializerExpr() . . . 123
5.1.1.113 ktc_is_Initializers() . . . 125
5.1.1.114 ktc_is_KRFuncDeclarator() . . . 125
5.1.1.115 ktc_is_Label() . . . 125
5.1.1.116 ktc_is_LabeledStmt() . . . 127
5.1.1.117 ktc_is_LambdaDeclarator() . . . 127
5.1.1.118 ktc_is_LambdaExpr() . . . 127
5.1.1.119 ktc_is_LambdaIntroducer() . . . 129
5.1.1.120 ktc_is_LeaveStmt() . . . 129
5.1.1.121 ktc_is_LinkageSpec() . . . 129
5.1.1.122 ktc_is_LiteralExpr() . . . 131
5.1.1.123 ktc_is_MaybeCtorInitializer() . . . 131
5.1.1.124 ktc_is_MaybeDeclarator() . . . 131
5.1.1.125 ktc_is_MaybeException() . . . 133
5.1.1.126 ktc_is_MaybeExceptionSpec() . . . 133
5.1.1.127 ktc_is_MaybeLambdaDeclarator() . . . 133
5.1.1.128 ktc_is_MaybeNewInitializer() . . . 135
5.1.1.129 ktc_is_MaybeTypeId() . . . 135
5.1.1.130 ktc_is_MemberDecl() . . . 135
5.1.1.131 ktc_is_MemberDecls() . . . 137
5.1.1.132 ktc_is_MemberDesignator() . . . 137
5.1.1.133 ktc_is_MemberExpr() . . . 137
5.1.1.134 ktc_is_MemberFunc() . . . 139
5.1.1.135 ktc_is_MemberInitializer() . . . 139
5.1.1.136 ktc_is_MemberInitializers() . . . 139

5.1.1.137 ktc_is_MemberTemplate() . . . 141
5.1.1.138 ktc_is_MemberUsingDecl() . . . 141
5.1.1.139 ktc_is_Name() . . . 141
5.1.1.140 ktc_is_NameDeclarator() . . . 143
5.1.1.141 ktc_is_NamespaceAlias() . . . 143
5.1.1.142 ktc_is_NamespaceDecl() . . . 143
5.1.1.143 ktc_is_NameSpec() . . . 145
5.1.1.144 ktc_is_NewExpr() . . . 145
5.1.1.145 ktc_is_NewInitializer() . . . 145
5.1.1.146 ktc_is_NoAttribute() . . . 147
5.1.1.147 ktc_is_NoAttributeSpec() . . . 147
5.1.1.148 ktc_is_NoBaseSpec() . . . 147
5.1.1.149 ktc_is_NoCapture() . . . 149
5.1.1.150 ktc_is_NoCtorInitializer() . . . 149
5.1.1.151 ktc_is_Node() . . . 149
5.1.1.152 ktc_is_NoDeclarator() . . . 151
5.1.1.153 ktc_is_NoDeclOrStmt() . . . 151
5.1.1.154 ktc_is_NoDeclSpec() . . . 151
5.1.1.155 ktc_is_NoDesignator() . . . 153
5.1.1.156 ktc_is_NoEnumerator() . . . 153
5.1.1.157 ktc_is_NoException() . . . 153
5.1.1.158 ktc_is_NoExceptionSpec() . . . 155
5.1.1.159 ktc_is_NoExpr() . . . 155
5.1.1.160 ktc_is_NoHandler() . . . 155
5.1.1.161 ktc_is_NoInitializer() . . . 157
5.1.1.162 ktc_is_NoLambdaDeclarator() . . . 157
5.1.1.163 ktc_is_NoMemberDecl() . . . 157
5.1.1.164 ktc_is_NoMemberInitializer() . . . 159
5.1.1.165 ktc_is_NoName() . . . 159
5.1.1.166 ktc_is_NoNameQualifier() . . . 159
5.1.1.167 ktc_is_NoNewInitializer() . . . 161
5.1.1.168 ktc_is_NoParamName() . . . 161
5.1.1.169 ktc_is_NoPropertyFunc() . . . 161
5.1.1.170 ktc_is_NoTemplateArg() . . . 163
5.1.1.171 ktc_is_NoTemplateParam() . . . 163
5.1.1.172 ktc_is_NoTypeId() . . . 163
5.1.1.173 ktc_is_NullptrLiteralExpr() . . . 165
5.1.1.174 ktc_is_OpFunc() . . . 165
5.1.1.175 ktc_is_Param() . . . 165
5.1.1.176 ktc_is_ParamName() . . . 167
5.1.1.177 ktc_is_ParamNames() . . . 167
5.1.1.178 ktc_is_ParensDeclarator() . . . 167

5.1.1.179 ktc_is_ParensExpr() . . . 169
5.1.1.180 ktc_is_PromisedFuncBody() . . . 169
5.1.1.181 ktc_is_PromisedMemberDecl() . . . 169
5.1.1.182 ktc_is_PropertyAttribute() . . . 171
5.1.1.183 ktc_is_PropertyFuncs() . . . 171
5.1.1.184 ktc_is_PropertyGetFunc() . . . 171
5.1.1.185 ktc_is_PropertyPutFunc() . . . 173
5.1.1.186 ktc_is_PseudoDtor() . . . 173
5.1.1.187 ktc_is_PtrDeclarator() . . . 173
5.1.1.188 ktc_is_QualifiedName() . . . 175
5.1.1.189 ktc_is_QualifiedPseudoDtor() . . . 175
5.1.1.190 ktc_is_RangeDesignator() . . . 175
5.1.1.191 ktc_is_ReservedTypeSpec() . . . 177
5.1.1.192 ktc_is_ReturnStmt() . . . 177
5.1.1.193 ktc_is_SizeOfExpr() . . . 177
5.1.1.194 ktc_is_SpecialCastExpr() . . . 179
5.1.1.195 ktc_is_StaticAssertDecl() . . . 179
5.1.1.196 ktc_is_StmtExpr() . . . 179
5.1.1.197 ktc_is_StorageClass() . . . 181
5.1.1.198 ktc_is_StringLiteralExpr() . . . 181
5.1.1.199 ktc_is_SuffixFunc() . . . 181
5.1.1.200 ktc_is_SuperScope() . . . 183
5.1.1.201 ktc_is_SwitchDeclStmt() . . . 183
5.1.1.202 ktc_is_SwitchStmt() . . . 183
5.1.1.203 ktc_is_TemplateArgs() . . . 185
5.1.1.204 ktc_is_TemplateDecl() . . . 185
5.1.1.205 ktc_is_TemplateName() . . . 185
5.1.1.206 ktc_is_TemplateParam() . . . 187
5.1.1.207 ktc_is_TemplateParams() . . . 187
5.1.1.208 ktc_is_TemplateSpec() . . . 187
5.1.1.209 ktc_is_TemplateTypeArg() . . . 189
5.1.1.210 ktc_is_TemplateTypeParam() . . . 189
5.1.1.211 ktc_is_ThisExpr() . . . 189
5.1.1.212 ktc_is_ThrowExpr() . . . 191
5.1.1.213 ktc_is_TranslationUnit() . . . 191
5.1.1.214 ktc_is_TruncatedInitClause() . . . 191
5.1.1.215 ktc_is_TryExceptStmt() . . . 193
5.1.1.216 ktc_is_TryFinallyStmt() . . . 193
5.1.1.217 ktc_is_TryStmt() . . . 193
5.1.1.218 ktc_is_TypeAdjective() . . . 195
5.1.1.219 ktc_is_TypeArg() . . . 195
5.1.1.220 ktc_is_TypeConvExpr() . . . 195

5.1.1.221 ktc_is_TypeId() . . . 197
5.1.1.222 ktc_is_TypeName() . . . 197
5.1.1.223 ktc_is_TypeOfExpr() . . . 197
5.1.1.224 ktc_is_TypeOfSpec() . . . 199
5.1.1.225 ktc_is_TypeOfType() . . . 199
5.1.1.226 ktc_is_TypeParam() . . . 199
5.1.1.227 ktc_is_TypeTypeIdExpr() . . . 201
5.1.1.228 ktc_is_UnaryExpr() . . . 201
5.1.1.229 ktc_is_UnparsedDecl() . . . 201
5.1.1.230 ktc_is_UnparsedDeclarator() . . . 203
5.1.1.231 ktc_is_UnparsedDeclSpec() . . . 203
5.1.1.232 ktc_is_UnparsedEnumerator() . . . 203
5.1.1.233 ktc_is_UnparsedException() . . . 205
5.1.1.234 ktc_is_UnparsedExpr() . . . 205
5.1.1.235 ktc_is_UnparsedInitializer() . . . 205
5.1.1.236 ktc_is_UnparsedLabel() . . . 207
5.1.1.237 ktc_is_UnparsedMemberDecl() . . . 207
5.1.1.238 ktc_is_UnparsedName() . . . 207
5.1.1.239 ktc_is_UnparsedNameQualifier() . . . 209
5.1.1.240 ktc_is_UnparsedParamName() . . . 209
5.1.1.241 ktc_is_UnparsedPropertyFunc() . . . 209
5.1.1.242 ktc_is_UnparsedStmt() . . . 211
5.1.1.243 ktc_is_UnqualifiedName() . . . 211
5.1.1.244 ktc_is_UserLiteralExpr() . . . 211
5.1.1.245 ktc_is_UserStringLiteralExpr() . . . 213
5.1.1.246 ktc_is_UsingDecl() . . . 213
5.1.1.247 ktc_is_UsingDirective() . . . 213
5.1.1.248 ktc_is_WhileDeclStmt() . . . 215
5.1.1.249 ktc_is_WhileStmt() . . . 215
5.1.2 Variable Documentation . . . 215
5.1.2.1 cid_Adjacent . . . 216
5.1.2.2 cid_Args . . . 216
5.1.2.3 cid_Attributes . . . 216
5.1.2.4 cid_AttributeSpec . . . 216
5.1.2.5 cid_AttributeSpecs . . . 216
5.1.2.6 cid_Base . . . 216
5.1.2.7 cid_BaseSpecs . . . 216
5.1.2.8 cid_Bits . . . 216
5.1.2.9 cid_Cond . . . 217
5.1.2.10 cid_ConversionType . . . 217
5.1.2.11 cid_CtorInit . . . 217
5.1.2.12 cid_CVQualifiers . . . 217

5.1.2.13 cid_Decl . . . 217
5.1.2.14 cid_Declarator . . . 217
5.1.2.15 cid_Declarators . . . 217
5.1.2.16 cid_Decls . . . 217
5.1.2.17 cid_DeclSpecs . . . 218
5.1.2.18 cid_Default . . . 218
5.1.2.19 cid_Designators . . . 218
5.1.2.20 cid_Else . . . 218
5.1.2.21 cid_Enumerators . . . 218
5.1.2.22 cid_Exception . . . 218
5.1.2.23 cid_Expr . . . 218
5.1.2.24 cid_Func . . . 218
5.1.2.25 cid_FuncBody . . . 219
5.1.2.26 cid_Handler . . . 219
5.1.2.27 cid_Handlers . . . 219
5.1.2.28 cid_Index . . . 219
5.1.2.29 cid_Init . . . 219
5.1.2.30 cid_Initializer . . . 219
5.1.2.31 cid_Inits . . . 219
5.1.2.32 cid_Introducer . . . 219
5.1.2.33 cid_KRParams . . . 220
5.1.2.34 cid_Label . . . 220
5.1.2.35 cid_LambdaCapture . . . 220
5.1.2.36 cid_Left . . . 220
5.1.2.37 cid_Literal . . . 220
5.1.2.38 cid_Lower . . . 220
5.1.2.39 cid_MemberDecl . . . 220
5.1.2.40 cid_MemberDecls . . . 220
5.1.2.41 cid_MemberInitializers . . . 221
5.1.2.42 cid_Name . . . 221
5.1.2.43 cid_NameSpec . . . 221
5.1.2.44 cid_Next . . . 221
5.1.2.45 cid_Params . . . 221
5.1.2.46 cid_Placement . . . 221
5.1.2.47 cid_PropertyFuncs . . . 221
5.1.2.48 cid_Qualifier . . . 221
5.1.2.49 cid_Right . . . 222
5.1.2.50 cid_Size . . . 222
5.1.2.51 cid_Stmt . . . 222
5.1.2.52 cid_Stmts . . . 222
5.1.2.53 cid_TemplateName . . . 222
5.1.2.54 cid_TemplateParams . . . 222

5.1.2.55 cid_Then . . . 222
5.1.2.56 cid_Throw . . . 222
5.1.2.57 cid_TrailingReturnType . . . 223
5.1.2.58 cid_Type . . . 223
5.1.2.59 cid_TypeIds . . . 223
5.1.2.60 cid_Upper . . . 223
5.1.2.61 tid_AccessSpecification . . . 223
5.1.2.62 tid_AliasDecl . . . 223
5.1.2.63 tid_AlignAsExpr . . . 223
5.1.2.64 tid_AlignAsType . . . 223
5.1.2.65 tid_AlignOfExpr . . . 224
5.1.2.66 tid_Any . . . 224
5.1.2.67 tid_AnyAttribute . . . 224
5.1.2.68 tid_AnyCapture . . . 224
5.1.2.69 tid_AnyDecl . . . 224
5.1.2.70 tid_AnyDeclarator . . . 224
5.1.2.71 tid_AnyDesignator . . . 224
5.1.2.72 tid_AnyEnumerator . . . 224
5.1.2.73 tid_AnyExpr . . . 225
5.1.2.74 tid_AnyFuncBody . . . 225
5.1.2.75 tid_AnyInitializer . . . 225
5.1.2.76 tid_AnyLabel . . . 225
5.1.2.77 tid_AnyMemberDecl . . . 225
5.1.2.78 tid_AnyName . . . 225
5.1.2.79 tid_AnyNameQualifier . . . 225
5.1.2.80 tid_AnyNames . . . 225
5.1.2.81 tid_AnyNameSpec . . . 226
5.1.2.82 tid_AnyNonPtrDeclarator . . . 226
5.1.2.83 tid_AnyParamName . . . 226
5.1.2.84 tid_AnyPropertyFunc . . . 226
5.1.2.85 tid_AnyPseudoDtor . . . 226
5.1.2.86 tid_AnyStmt . . . 226
5.1.2.87 tid_AnyTemplateArg . . . 226
5.1.2.88 tid_AnyTypeName . . . 226
5.1.2.89 tid_AnyTypeOf . . . 227
5.1.2.90 tid_AnyTypeParam . . . 227
5.1.2.91 tid_AnyUsing . . . 227
5.1.2.92 tid_ArrayDeclarator . . . 227
5.1.2.93 tid_AsmDef . . . 227
5.1.2.94 tid_AsmStmt . . . 227
5.1.2.95 tid_Attribute . . . 227
5.1.2.96 tid_AttributedDeclarator . . . 227

5.1.2.97 tid_AttributeDeclSpec . . . 228
5.1.2.98 tid_Attributes . . . 228
5.1.2.99 tid_AttributeSpec . . . 228
5.1.2.100 tid_AttributeSpecs . . . 228
5.1.2.101 tid_AttributeWithArgs . . . 228
5.1.2.102 tid_AutoType . . . 228
5.1.2.103 tid_BaseSpec . . . 228
5.1.2.104 tid_BaseSpecs . . . 228
5.1.2.105 tid_BinaryExpr . . . 229
5.1.2.106 tid_BitFieldDeclarator . . . 229
5.1.2.107 tid_BoolLiteralExpr . . . 229
5.1.2.108 tid_BreakStmt . . . 229
5.1.2.109 tid_BuiltinType . . . 229
5.1.2.110 tid_CallExpr . . . 229
5.1.2.111 tid_Capture . . . 229
5.1.2.112 tid_CaptureDefault . . . 229
5.1.2.113 tid_CaseLabel . . . 230
5.1.2.114 tid_CaseRangeLabel . . . 230
5.1.2.115 tid_CastExpr . . . 230
5.1.2.116 tid_ClassType . . . 230
5.1.2.117 tid_CompoundStmt . . . 230
5.1.2.118 tid_ConditionalExpr . . . 230
5.1.2.119 tid_ConstExpr . . . 230
5.1.2.120 tid_ContinueStmt . . . 230
5.1.2.121 tid_ConvFunc . . . 231
5.1.2.122 tid_CopyInitializer . . . 231
5.1.2.123 tid_CtorInitializer . . . 231
5.1.2.124 tid_CVQualifier . . . 231
5.1.2.125 tid_Decl . . . 231
5.1.2.126 tid_DeclEllipsis . . . 231
5.1.2.127 tid_DeclOrStmt . . . 231
5.1.2.128 tid_DeclOrStmts . . . 231
5.1.2.129 tid_DeclSpec . . . 232
5.1.2.130 tid_DeclSpecs . . . 232
5.1.2.131 tid_DefaultException . . . 232
5.1.2.132 tid_DefaultLabel . . . 232
5.1.2.133 tid_DeleteExpr . . . 232
5.1.2.134 tid_DenyThrowSpec . . . 232
5.1.2.135 tid_Designators . . . 232
5.1.2.136 tid_DirectInitializer . . . 232
5.1.2.137 tid_DoDeclStmt . . . 233
5.1.2.138 tid_DoStmt . . . 233

5.1.2.139 tid_Dtor . . . 233
5.1.2.140 tid_Enumerator . . . 233
5.1.2.141 tid_Enumerators . . . 233
5.1.2.142 tid_EnumType . . . 233
5.1.2.143 tid_ExceptHandler . . . 233
5.1.2.144 tid_Exception . . . 233
5.1.2.145 tid_ExceptionSpec . . . 234
5.1.2.146 tid_ExplicitInstantiation . . . 234
5.1.2.147 tid_ExprArg . . . 234
5.1.2.148 tid_Exprs . . . 234
5.1.2.149 tid_ExprStmt . . . 234
5.1.2.150 tid_ExprTypeIdExpr . . . 234
5.1.2.151 tid_FieldDesignator . . . 234
5.1.2.152 tid_FinallyHandler . . . 234
5.1.2.153 tid_ForEachStmt . . . 235
5.1.2.154 tid_ForRangeStmt . . . 235
5.1.2.155 tid_ForStmt . . . 235
5.1.2.156 tid_FuncBody . . . 235
5.1.2.157 tid_FuncDeclarator . . . 235
5.1.2.158 tid_FuncDef . . . 235
5.1.2.159 tid_FuncSpec . . . 235
5.1.2.160 tid_FuncTryBlock . . . 235
5.1.2.161 tid_GenericAttribute . . . 236
5.1.2.162 tid_GlobalScope . . . 236
5.1.2.163 tid_GotoStmt . . . 236
5.1.2.164 tid_Handler . . . 236
5.1.2.165 tid_Handlers . . . 236
5.1.2.166 tid_IdExpr . . . 236
5.1.2.167 tid_IfDeclStmt . . . 236
5.1.2.168 tid_IfStmt . . . 236
5.1.2.169 tid_IndexDesignator . . . 237
5.1.2.170 tid_IndexExpr . . . 237
5.1.2.171 tid_InitClause . . . 237
5.1.2.172 tid_InitializedDeclarator . . . 237
5.1.2.173 tid_InitializerExpr . . . 237
5.1.2.174 tid_Initializers . . . 237
5.1.2.175 tid_KRFuncDeclarator . . . 237
5.1.2.176 tid_Label . . . 237
5.1.2.177 tid_LabeledStmt . . . 238
5.1.2.178 tid_LambdaDeclarator . . . 238
5.1.2.179 tid_LambdaExpr . . . 238
5.1.2.180 tid_LambdaIntroducer . . . 238

5.1.2.181 tid_LeaveStmt . . . 238
5.1.2.182 tid_LinkageSpec . . . 238
5.1.2.183 tid_LiteralExpr . . . 238
5.1.2.184 tid_MaybeCtorInitializer . . . 238
5.1.2.185 tid_MaybeDeclarator . . . 239
5.1.2.186 tid_MaybeException . . . 239
5.1.2.187 tid_MaybeExceptionSpec . . . 239
5.1.2.188 tid_MaybeLambdaDeclarator . . . 239
5.1.2.189 tid_MaybeNewInitializer . . . 239
5.1.2.190 tid_MaybeTypeId . . . 239
5.1.2.191 tid_MemberDecl . . . 239
5.1.2.192 tid_MemberDecls . . . 239
5.1.2.193 tid_MemberDesignator . . . 240
5.1.2.194 tid_MemberExpr . . . 240
5.1.2.195 tid_MemberFunc . . . 240
5.1.2.196 tid_MemberInitializer . . . 240
5.1.2.197 tid_MemberInitializers . . . 240
5.1.2.198 tid_MemberTemplate . . . 240
5.1.2.199 tid_MemberUsingDecl . . . 240
5.1.2.200 tid_Name . . . 240
5.1.2.201 tid_NameDeclarator . . . 241
5.1.2.202 tid_NamespaceAlias . . . 241
5.1.2.203 tid_NamespaceDecl . . . 241
5.1.2.204 tid_NameSpec . . . 241
5.1.2.205 tid_NewExpr . . . 241
5.1.2.206 tid_NewInitializer . . . 241
5.1.2.207 tid_NoAttribute . . . 241
5.1.2.208 tid_NoAttributeSpec . . . 241
5.1.2.209 tid_NoBaseSpec . . . 242
5.1.2.210 tid_NoCapture . . . 242
5.1.2.211 tid_NoCtorInitializer . . . 242
5.1.2.212 tid_Node . . . 242
5.1.2.213 tid_NoDeclarator . . . 242
5.1.2.214 tid_NoDeclOrStmt . . . 242
5.1.2.215 tid_NoDeclSpec . . . 242
5.1.2.216 tid_NoDesignator . . . 242
5.1.2.217 tid_NoEnumerator . . . 243
5.1.2.218 tid_NoException . . . 243
5.1.2.219 tid_NoExceptionSpec . . . 243
5.1.2.220 tid_NoExpr . . . 243
5.1.2.221 tid_NoHandler . . . 243
5.1.2.222 tid_NoInitializer . . . 243

5.1.2.223 tid_NoLambdaDeclarator . . . 243
5.1.2.224 tid_NoMemberDecl . . . 243
5.1.2.225 tid_NoMemberInitializer . . . 244
5.1.2.226 tid_NoName . . . 244
5.1.2.227 tid_NoNameQualifier . . . 244
5.1.2.228 tid_NoNewInitializer . . . 244
5.1.2.229 tid_NoParamName . . . 244
5.1.2.230 tid_NoPropertyFunc . . . 244
5.1.2.231 tid_NoTemplateArg . . . 244
5.1.2.232 tid_NoTemplateParam . . . 244
5.1.2.233 tid_NoTypeId . . . 245
5.1.2.234 tid_NullptrLiteralExpr . . . 245
5.1.2.235 tid_OpFunc . . . 245
5.1.2.236 tid_Param . . . 245
5.1.2.237 tid_ParamName . . . 245
5.1.2.238 tid_ParamNames . . . 245
5.1.2.239 tid_ParensDeclarator . . . 245
5.1.2.240 tid_ParensExpr . . . 245
5.1.2.241 tid_PromisedFuncBody . . . 246
5.1.2.242 tid_PromisedMemberDecl . . . 246
5.1.2.243 tid_PropertyAttribute . . . 246
5.1.2.244 tid_PropertyFuncs . . . 246
5.1.2.245 tid_PropertyGetFunc . . . 246
5.1.2.246 tid_PropertyPutFunc . . . 246
5.1.2.247 tid_PseudoDtor . . . 246
5.1.2.248 tid_PtrDeclarator . . . 246
5.1.2.249 tid_QualifiedName . . . 247
5.1.2.250 tid_QualifiedPseudoDtor . . . 247
5.1.2.251 tid_RangeDesignator . . . 247
5.1.2.252 tid_ReservedTypeSpec . . . 247
5.1.2.253 tid_ReturnStmt . . . 247
5.1.2.254 tid_SizeOfExpr . . . 247
5.1.2.255 tid_SpecialCastExpr . . . 247
5.1.2.256 tid_StaticAssertDecl . . . 247
5.1.2.257 tid_StmtExpr . . . 248
5.1.2.258 tid_StorageClass . . . 248
5.1.2.259 tid_StringLiteralExpr . . . 248
5.1.2.260 tid_SuffixFunc . . . 248
5.1.2.261 tid_SuperScope . . . 248
5.1.2.262 tid_SwitchDeclStmt . . . 248
5.1.2.263 tid_SwitchStmt . . . 248
5.1.2.264 tid_TemplateArgs . . . 248

5.1.2.265 tid_TemplateDecl . . . 249
5.1.2.266 tid_TemplateName . . . 249
5.1.2.267 tid_TemplateParam . . . 249
5.1.2.268 tid_TemplateParams . . . 249
5.1.2.269 tid_TemplateSpec . . . 249
5.1.2.270 tid_TemplateTypeArg . . . 249
5.1.2.271 tid_TemplateTypeParam . . . 249
5.1.2.272 tid_ThisExpr . . . 249
5.1.2.273 tid_ThrowExpr . . . 250
5.1.2.274 tid_TranslationUnit . . . 250
5.1.2.275 tid_TruncatedInitClause . . . 250
5.1.2.276 tid_TryExceptStmt . . . 250
5.1.2.277 tid_TryFinallyStmt . . . 250
5.1.2.278 tid_TryStmt . . . 250
5.1.2.279 tid_TypeAdjective . . . 250
5.1.2.280 tid_TypeArg . . . 250
5.1.2.281 tid_TypeConvExpr . . . 251
5.1.2.282 tid_TypeId . . . 251
5.1.2.283 tid_TypeName . . . 251
5.1.2.284 tid_TypeOfExpr . . . 251
5.1.2.285 tid_TypeOfSpec . . . 251
5.1.2.286 tid_TypeOfType . . . 251
5.1.2.287 tid_TypeParam . . . 251
5.1.2.288 tid_TypeTypeIdExpr . . . 251
5.1.2.289 tid_UnaryExpr . . . 252
5.1.2.290 tid_UnparsedDecl . . . 252
5.1.2.291 tid_UnparsedDeclarator . . . 252
5.1.2.292 tid_UnparsedDeclSpec . . . 252
5.1.2.293 tid_UnparsedEnumerator . . . 252
5.1.2.294 tid_UnparsedException . . . 252
5.1.2.295 tid_UnparsedExpr . . . 252
5.1.2.296 tid_UnparsedInitializer . . . 252
5.1.2.297 tid_UnparsedLabel . . . 253
5.1.2.298 tid_UnparsedMemberDecl . . . 253
5.1.2.299 tid_UnparsedName . . . 253
5.1.2.300 tid_UnparsedNameQualifier . . . 253
5.1.2.301 tid_UnparsedParamName . . . 253
5.1.2.302 tid_UnparsedPropertyFunc . . . 253
5.1.2.303 tid_UnparsedStmt . . . 253
5.1.2.304 tid_UnqualifiedName . . . 253
5.1.2.305 tid_UserLiteralExpr . . . 254
5.1.2.306 tid_UserStringLiteralExpr . . . 254

5.1.2.307 tid_UsingDecl . . . 254
5.1.2.308 tid_UsingDirective . . . 254
5.1.2.309 tid_WhileDeclStmt . . . 254
5.1.2.310 tid_WhileStmt . . . 254
5.2 ktcAPI.h File Reference . . . 254
5.3 ktcMainAPI.h File Reference . . . 255
5.3.1 Macro Definition Documentation . . . 260
5.3.1.1 KTC_API_VERSION_MAJOR . . . 260
5.3.1.2 KTC_API_VERSION_MINOR . . . 260
5.3.1.3 KTC_API_VERSION_PATCHLEVEL . . . 260
5.3.1.4 KTC_CUSTOM_TYPES . . . 260
5.3.1.5 ktc_require . . . 260
5.3.2 Typedef Documentation . . . 260
5.3.2.1 ktc_autofix_t . . . 261
5.3.2.2 ktc_childId_t . . . 261
5.3.2.3 ktc_eventHook_t . . . 261
5.3.2.4 ktc_languageType_t . . . 261
5.3.2.5 ktc_long_long_t . . . 261
5.3.2.6 ktc_message . . . 261
5.3.2.7 ktc_message_t . . . 261
5.3.2.8 ktc_position . . . 262
5.3.2.9 ktc_position_t . . . 262
5.3.2.10 ktc_semanticInfo_t . . . 262
5.3.2.11 ktc_string_t . . . 262
5.3.2.12 ktc_tree_t . . . 262
5.3.2.13 ktc_treeHook_t . . . 262
5.3.2.14 ktc_treeType_t . . . 262
5.3.3 Function Documentation . . . 262
5.3.3.1 ktc_assembleStringConstant() . . . 263
5.3.3.2 ktc_autofix_addSegment() . . . 264
5.3.3.3 ktc_autofix_delete() . . . 264
5.3.3.4 ktc_autofix_new() . . . 264
5.3.3.5 ktc_compareSubtrees() . . . 265
5.3.3.6 ktc_error_getConfigurationParameter() . . . 265
5.3.3.7 ktc_error_isEnabled() . . . 265
5.3.3.8 ktc_event_new() . . . 266
5.3.3.9 ktc_event_setParameter() . . . 266
5.3.3.10 ktc_forAllSubtreeNodes() . . . 266
5.3.3.11 ktc_free() . . . 266
5.3.3.12 ktc_getAssociatedScope() . . . 267
5.3.3.13 ktc_getBuiltinType() . . . 267
5.3.3.14 ktc_getBuiltinTypeSize() . . . 267

5.3.3.15 ktc_getCallArgument() . . . 268
5.3.3.16 ktc_getCalledFunction() . . . 268
5.3.3.17 ktc_getCastSpecifier() . . . 268
5.3.3.18 ktc_getClassTag() . . . 269
5.3.3.19 ktc_getEndPosition() . . . 269
5.3.3.20 ktc_getFrontendDialect() . . . 269
5.3.3.21 ktc_getFrontendLanguage() . . . 270
5.3.3.22 ktc_getFunctionSpecifier() . . . 270
5.3.3.23 ktc_getIdentifier() . . . 270
5.3.3.24 ktc_getIdentifierNo() . . . 270
5.3.3.25 ktc_getIntegerValue() . . . 270
5.3.3.26 ktc_getLanguageType() . . . 271
5.3.3.27 ktc_getNameDeclarator() . . . 271
5.3.3.28 ktc_getNoIdent() . . . 271
5.3.3.29 ktc_getNumberOfCallArguments() . . . 271
5.3.3.30 ktc_getOperation() . . . 272
5.3.3.31 ktc_getOutputFileName() . . . 272
5.3.3.32 ktc_getPointedType() . . . 272
5.3.3.33 ktc_getPointerOperator() . . . 273
5.3.3.34 ktc_getPointerSize() . . . 273
5.3.3.35 ktc_getSemanticInfo() . . . 273
5.3.3.36 ktc_getSizeofArgument() . . . 273
5.3.3.37 ktc_getStartPosition() . . . 273
5.3.3.38 ktc_getStorageClass() . . . 274
5.3.3.39 ktc_getStringConstantValue() . . . 274
5.3.3.40 ktc_getTokens() . . . 274
5.3.3.41 ktc_getTypeQualifiers() . . . 274
5.3.3.42 ktc_hasBuiltinWideChar() . . . 275
5.3.3.43 ktc_is_NoToken() . . . 275
5.3.3.44 ktc_is_Token() . . . 275
5.3.3.45 ktc_isCallTo() . . . 276
5.3.3.46 ktc_isCharLiteral() . . . 276
5.3.3.47 ktc_isConstructor() . . . 276
5.3.3.48 ktc_isDeclaration() . . . 277
5.3.3.49 ktc_isDefinition() . . . 277
5.3.3.50 ktc_isIncluded() . . . 277
5.3.3.51 ktc_isMacroExpansion() . . . 277
5.3.3.52 ktc_isMacroExpansion2() . . . 278
5.3.3.53 ktc_isNameIncluded() . . . 278
5.3.3.54 ktc_isNullMacro() . . . 278
5.3.3.55 ktc_isNullPointerConstant() . . . 279
5.3.3.56 ktc_isOperationOverloaded() . . . 279

5.3.3.57 ktc_isStatic() . . . 279
5.3.3.58 ktc_isTreeType() . . . 280
5.3.3.59 ktc_isUTF16String() . . . 281
5.3.3.60 ktc_isUTF32String() . . . 281
5.3.3.61 ktc_isWideString() . . . 282
5.3.3.62 ktc_languageTypeEffectiveSignedness() . . . 282
5.3.3.63 ktc_languageTypeIsBuiltin() . . . 282
5.3.3.64 ktc_languageTypeIsPointer() . . . 282
5.3.3.65 ktc_languageTypeSignedness() . . . 283
5.3.3.66 ktc_languageTypeSize() . . . 283
5.3.3.67 ktc_message_addAnchorAttribute() . . . 283
5.3.3.68 ktc_message_addAttribute() . . . 283
5.3.3.69 ktc_message_addEvent() . . . 284
5.3.3.70 ktc_message_addEventEx() . . . 284
5.3.3.71 ktc_message_addTraceBySemanticsInfo() . . . 284
5.3.3.72 ktc_message_delete() . . . 284
5.3.3.73 ktc_message_new() . . . 284
5.3.3.74 ktc_message_render() . . . 285
5.3.3.75 ktc_message_render_wi_autofix() . . . 285
5.3.3.76 ktc_message_setFunction() . . . 285
5.3.3.77 ktc_message_setPosition() . . . 285
5.3.3.78 ktc_message_setRecommendationFactor() . . . 286
5.3.3.79 ktc_message_unsetFunction() . . . 286
5.3.3.80 ktc_nodeStackGet() . . . 286
5.3.3.81 ktc_nodeStackTop() . . . 286
5.3.3.82 ktc_position_copy() . . . 287
5.3.3.83 ktc_position_delete() . . . 287
5.3.3.84 ktc_position_getColumn() . . . 287
5.3.3.85 ktc_position_getFileName() . . . 287
5.3.3.86 ktc_position_getLine() . . . 287
5.3.3.87 ktc_position_new() . . . 287
5.3.3.88 ktc_position_setLine() . . . 288
5.3.3.89 ktc_proceed() . . . 288
5.3.3.90 ktc_registerRestoreContextHook() . . . 288
5.3.3.91 ktc_registerSaveContextHook() . . . 289
5.3.3.92 ktc_registerStartTraverseHook() . . . 289
5.3.3.93 ktc_registerStopTraverseHook() . . . 289
5.3.3.94 ktc_registerTreeHook() . . . 289
5.3.3.95 ktc_sema_checkBitField() . . . 289
5.3.3.96 ktc_sema_findAllByName() . . . 290
5.3.3.97 ktc_sema_findFirstByName() . . . 290
5.3.3.98 ktc_sema_forAllClassDeclarations() . . . 290

5.3.3.99 ktc_sema_forAllScopeDeclarations() . . . 291
5.3.3.100 ktc_sema_forAllSubtreeNodes() . . . 291
5.3.3.101 ktc_sema_functionOverloadsFunction() . . . 291
5.3.3.102 ktc_sema_getAllByName() . . . 292
5.3.3.103 ktc_sema_getArrayElementType() . . . 292
5.3.3.104 ktc_sema_getArraySize() . . . 292
5.3.3.105 ktc_sema_getAST() . . . 293
5.3.3.106 ktc_sema_getBaseInfo() . . . 293
5.3.3.107 ktc_sema_getBuiltinCode() . . . 293
5.3.3.108 ktc_sema_getClassTag() . . . 294
5.3.3.109 ktc_sema_getCVQualifiers() . . . 294
5.3.3.110 ktc_sema_getDefinedType() . . . 294
5.3.3.111 ktc_sema_getEnumerators() . . . 294
5.3.3.112 ktc_sema_getEnumUnderlyingTypeOrInt() . . . 295
5.3.3.113 ktc_sema_getFieldNumber() . . . 295
5.3.3.114 ktc_sema_getFieldType() . . . 295
5.3.3.115 ktc_sema_getFirstByName() . . . 296
5.3.3.116 ktc_sema_getFormalArgument() . . . 296
5.3.3.117 ktc_sema_getFunctionFromTemplate() . . . 297
5.3.3.118 ktc_sema_getFunctionPointerType() . . . 297
5.3.3.119 ktc_sema_getFunctionTemplateInstantiations() . . . 297
5.3.3.120 ktc_sema_getFunctionTemplateSpecializations() . . . 297
5.3.3.121 ktc_sema_getFunctionType() . . . 297
5.3.3.122 ktc_sema_getGlobalScope() . . . 298
5.3.3.123 ktc_sema_getIdentifier() . . . 298
5.3.3.124 ktc_sema_getIdentifierNo() . . . 298
5.3.3.125 ktc_sema_getInstantiatedClass() . . . 298
5.3.3.126 ktc_sema_getInstantiationOrigin() . . . 298
5.3.3.127 ktc_sema_getInstantiationParameters() . . . 298
5.3.3.128 ktc_sema_getIntegerValue() . . . 299
5.3.3.129 ktc_sema_getIntValueType() . . . 299
5.3.3.130 ktc_sema_getNumber() . . . 300
5.3.3.131 ktc_sema_getNumberOfArguments() . . . 300
5.3.3.132 ktc_sema_getNumberOfBaseInfo() . . . 300
5.3.3.133 ktc_sema_getObjectName() . . . 300
5.3.3.134 ktc_sema_getOverridenMethod() . . . 301
5.3.3.135 ktc_sema_getPointedType() . . . 301
5.3.3.136 ktc_sema_getPosition() . . . 301
5.3.3.137 ktc_sema_getPrimaryTemplate() . . . 302
5.3.3.138 ktc_sema_getQualifiedName() . . . 302
5.3.3.139 ktc_sema_getReferencedType() . . . 302
5.3.3.140 ktc_sema_getReturnType() . . . 303

5.3.3.141 ktc_sema_getScope() . . . 303
5.3.3.142 ktc_sema_getTypedefedName() . . . 303
5.3.3.143 ktc_sema_getTypeName() . . . 303
5.3.3.144 ktc_sema_getVariableInitializer() . . . 304
5.3.3.145 ktc_sema_getVariableType() . . . 304
5.3.3.146 ktc_sema_getVariableValue() . . . 304
5.3.3.147 ktc_sema_hasMethods() . . . 304
5.3.3.148 ktc_sema_haveSameFunctionType() . . . 305
5.3.3.149 ktc_sema_haveSameSignature() . . . 305
5.3.3.150 ktc_sema_isAnonymous() . . . 305
5.3.3.151 ktc_sema_isArray() . . . 305
5.3.3.152 ktc_sema_isBasePrivate() . . . 305
5.3.3.153 ktc_sema_isBaseProtected() . . . 306
5.3.3.154 ktc_sema_isBasePublic() . . . 306
5.3.3.155 ktc_sema_isBaseVirtual() . . . 307
5.3.3.156 ktc_sema_isBitfield() . . . 307
5.3.3.157 ktc_sema_isBuiltin() . . . 307
5.3.3.158 ktc_sema_isClass() . . . 307
5.3.3.159 ktc_sema_isConstMethod() . . . 308
5.3.3.160 ktc_sema_isConstructor() . . . 308
5.3.3.161 ktc_sema_isDestructor() . . . 308
5.3.3.162 ktc_sema_isEnum() . . . 308
5.3.3.163 ktc_sema_isEnumConstant() . . . 308
5.3.3.164 ktc_sema_isFriend() . . . 309
5.3.3.165 ktc_sema_isFuncDef() . . . 309
5.3.3.166 ktc_sema_isFunction() . . . 309
5.3.3.167 ktc_sema_isFunctionPointer() . . . 309
5.3.3.168 ktc_sema_isFunctionTemplate() . . . 309
5.3.3.169 ktc_sema_isFunctionTemplateSet() . . . 309
5.3.3.170 ktc_sema_isFunctionType() . . . 310
5.3.3.171 ktc_sema_isGlobal() . . . 310
5.3.3.172 ktc_sema_isImmutable() . . . 310
5.3.3.173 ktc_sema_isIncluded() . . . 310
5.3.3.174 ktc_sema_isInline() . . . 310
5.3.3.175 ktc_sema_isInstantiatedFunction() . . . 311
5.3.3.176 ktc_sema_isInstantiation() . . . 311
5.3.3.177 ktc_sema_isIntegerValue() . . . 311
5.3.3.178 ktc_sema_isLocal() . . . 311
5.3.3.179 ktc_sema_isNamespace() . . . 311
5.3.3.180 ktc_sema_isNamespaceAlias() . . . 311
5.3.3.181 ktc_sema_isNone() . . . 311
5.3.3.182 ktc_sema_isObjectValue() . . . 312

5.3.3.183 ktc_sema_isOperatorFunction() . . . 312
5.3.3.184 ktc_sema_isPOD() . . . 312
5.3.3.185 ktc_sema_isPointer() . . . 312
5.3.3.186 ktc_sema_isPrivate() . . . 312
5.3.3.187 ktc_sema_isProtected() . . . 312
5.3.3.188 ktc_sema_isPublic() . . . 312
5.3.3.189 ktc_sema_isPureVirtual() . . . 313
5.3.3.190 ktc_sema_isReference() . . . 313
5.3.3.191 ktc_sema_isSameClass() . . . 313
5.3.3.192 ktc_sema_isSameEnum() . . . 313
5.3.3.193 ktc_sema_isSameFunctions() . . . 313
5.3.3.194 ktc_sema_isSameScope() . . . 314
5.3.3.195 ktc_sema_isSameVariables() . . . 314
5.3.3.196 ktc_sema_isScope() . . . 314
5.3.3.197 ktc_sema_isSpecialization() . . . 314
5.3.3.198 ktc_sema_isStatic() . . . 314
5.3.3.199 ktc_sema_isTemplate() . . . 315
5.3.3.200 ktc_sema_isType() . . . 315
5.3.3.201 ktc_sema_isTypeParameter() . . . 315
5.3.3.202 ktc_sema_isUnion() . . . 315
5.3.3.203 ktc_sema_isUsing() . . . 315
5.3.3.204 ktc_sema_isUsingDeclaration() . . . 315
5.3.3.205 ktc_sema_isUsingDirective() . . . 315
5.3.3.206 ktc_sema_isVariable() . . . 316
5.3.3.207 ktc_sema_isVirtual() . . . 316
5.3.3.208 ktc_sema_RelatedClasses() . . . 316
5.3.3.209 ktc_sema_skipTypedefs() . . . 316
5.3.3.210 ktc_sema_stripCVQ() . . . 317
5.3.3.211 ktc_skipBrackets() . . . 317
5.3.3.212 ktc_string_delete() . . . 317
5.3.3.213 ktc_string_get_cstring() . . . 317
5.3.3.214 ktc_string_new() . . . 318
5.3.3.215 ktc_treeType_getName() . . . 318
5.3.3.216 ktc_unregisterTreeHook() . . . 318
5.3.4 Variable Documentation . . . 318
5.3.4.1 ktc_constructorName . . . 319
5.3.4.2 ktc_destructorName . . . 319
5.3.4.3 KTC_DIALECT_ARM . . . 319
5.3.4.4 KTC_DIALECT_GHS . . . 319
5.3.4.5 KTC_DIALECT_GNU . . . 319
5.3.4.6 KTC_DIALECT_MS . . . 319
5.3.4.7 KTC_DIALECT_STD . . . 319

5.3.4.8 KTC_DIALECT_SUN . . . 319
5.3.4.9 KTC_DIALECT_TI . . . 320
5.3.4.10 KTC_LANGUAGE_C . . . 320
5.3.4.11 KTC_LANGUAGE_CXX . . . 320
5.3.4.12 KTC_TREE_EVENT_ON_ENTER . . . 320
5.3.4.13 KTC_TREE_EVENT_ON_LEAVE . . . 320
5.3.4.14 KTC_TREE_EVENT_ON_NEXT . . . 320
5.3.4.15 KTC_TYPESIGNEDNESS_DEFAULT . . . 320
5.3.4.16 KTC_TYPESIGNEDNESS_NONE . . . 321
5.3.4.17 KTC_TYPESIGNEDNESS_SIGNED . . . 321
5.3.4.18 KTC_TYPESIGNEDNESS_UNSIGNED . . . 321
5.4 kwapi.h File Reference . . . 321
5.4.1 Macro Definition Documentation . . . 322
5.4.1.1 KWAPI_DECLARE . . . 322
5.4.1.2 KWAPI_DECLARE_CPP . . . 322
5.4.1.3 KWAPI_DECLARE_DATA . . . 322
5.4.1.4 KWAPI_DECLARE_NONSTD . . . 322
5.4.2 Typedef Documentation . . . 323
5.4.2.1 kw_array_t . . . 323
5.4.2.2 kw_size_t . . . 323
5.4.2.3 kwapi_cfgparam_t . . . 323
5.4.3 Enumeration Type Documentation . . . 323
5.4.3.1 kwapi_apitypes_t . . . 323
5.4.3.2 kwapi_langtypes_t . . . 323
5.4.4 Function Documentation . . . 324
5.4.4.1 ktc_error_getConfigurationParameter() . . . 324
5.4.4.2 kw_array_delete() . . . 324
5.4.4.3 kw_array_get() . . . 324
5.4.4.4 kw_array_size() . . . 324
5.4.4.5 kwapi_cfgparam_errorIsEnabled() . . . 324
5.4.4.6 kwapi_cfgparam_getCheckerErrors() . . . 325
5.4.4.7 kwapi_cfgparam_getConfigurationParameter() . . . 325
5.4.4.8 kwapi_cfgparam_getListLength() . . . 325
5.4.4.9 kwapi_cfgparam_getListNodeByIndex() . . . 326
5.4.4.10 kwapi_cfgparam_getListNodeByName() . . . 326
5.4.4.11 kwapi_cfgparam_getListNodeByRegexMatchingName() . . . 326
5.4.4.12 kwapi_cfgparam_getName() . . . 326
5.4.4.13 kwapi_cfgparam_getParameterValue() . . . 327
5.4.4.14 kwapi_cfgparam_getParameterValueFromList() . . . 327
5.4.4.15 kwapi_cfgparam_getRootParameterList() . . . 327
5.4.4.16 kwapi_cfgparam_getType() . . . 327
5.4.4.17 kwapi_cfgparam_isParameter() . . . 327

Index 329

Chapter 1

Deprecated List

Member ktc_languageType_t (p. 261)

Use **ktc_semanticInfo_t** (p. 262) type

Member ktc_languageTypeIsPointer (p. 282) (ktc_languageType_t type)

Use **ktc_sema_isPointer** (p. 312)

Member ktc_getPointedType (p. 272) (ktc_languageType_t ptr_type)

Use **ktc_sema_getPointedType** (p. 301)

Member ktc_languageTypeIsBuiltin (p. 282) (ktc_languageType_t type, int builtin_code)

Use **ktc_sema_getBuiltinCode** (p. 293) instead

Member ktc_error_isEnabled (p. 265) (const char ∗error_id)

Use kwapi_cfgparam_errorIsEnabled

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Obtaining configuration parameters for an error . . . 7
Basic Abstract Syntax Tree traversal and checking routines . . . 8
Accessing node stack . . . 9
Setting and clearing handlers for events during tree traversal . . . 10
Access to semantic information . . . 11
Functions for accessing type information . . . 12
Attribute values for signedness of types . . . 13
Utility functions for tree access . . . 14
Working with tree positions . . . 15
for defect . . . 16
Accessing compiler configuration . . . 17
Working with warning and error messages . . . 18
Tree type identifiers . . . 19
Tree type checking predicates . . . 20
Child link identifiers . . . 21
Numerical codes of operations . . . 22
Numerical codes of declaration storage class specifiers . . . 30
Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFI↩ER_NONE and '&' for two other values . . . 32
Numerical codes to differentiate struct/class/union declarations . . . 33
Numerical codes of C/C++ built-in types . . . 34
Numerical codes for identifying inline/virtual/explicit/friend member function specifiers . 38
Numerical codes for identifying different C++ -style cast expressions . . . 39
Numerical codes for identifying declarator ptr types . . . 40

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

gen-ktcAPI.h 41
ktcAPI.h 254
ktcMainAPI.h 255
kwapi.h 321

Chapter 4

Module Documentation

4.1 Obtaining configuration parameters for an error

Checker configuration XML file may store parameters that checker may access using functions from this section.

Checker configuration XML file may store parameters that checker may access using functions from this section.

```
<error id="MYDEFECT" message="my message" severity="3" enabled="true">
  <parameter name="myparameter" value="yes"/>
</error>
```

4.2 Basic Abstract Syntax Tree traversal and checking routines

4.3 Accessing node stack

4.4 Setting and clearing handlers for events during tree traversal

4.5 Access to semantic information

4.6 Functions for accessing type information

4.7 Attribute values for signedness of types

4.8 Utility functions for tree access

4.9 Working with tree positions

4.10 for defect

4.11 Accessing compiler configuration

4.12 Working with warning and error messages

4.13 Tree type identifiers

This header file is generated by a plug-in. Do not do any changes here as they will be lost.

4.14 Tree type checking predicates

4.15 Child link identifiers

4.16 Numerical codes of operations

Modules

- **Numerical codes of declaration storage class specifiers**

Variables

- const int **KTC_OPCODE_NONE**
- const int **KTC_OPCODE_ASSIGN**
- const int **KTC_OPCODE_MULASSIGN**
- const int **KTC_OPCODE_DIVASSIGN**
- const int **KTC_OPCODE_MODASSIGN**
- const int **KTC_OPCODE_ADDASSIGN**
- const int **KTC_OPCODE_SUBASSIGN**
- const int **KTC_OPCODE_ASLASSIGN**
- const int **KTC_OPCODE_ASRASSIGN**
- const int **KTC_OPCODE_ANDASSIGN**
- const int **KTC_OPCODE_XORASSIGN**
- const int **KTC_OPCODE_ORASSIGN**
- const int **KTC_OPCODE_COMMA**
- const int **KTC_OPCODE_COND**
- const int **KTC_OPCODE_LOGOR**
- const int **KTC_OPCODE_LOGAND**
- const int **KTC_OPCODE_BITOR**
- const int **KTC_OPCODE_BITXOR**
- const int **KTC_OPCODE_BITAND**
- const int **KTC_OPCODE_EQ**
- const int **KTC_OPCODE_NE**
- const int **KTC_OPCODE_LT**
- const int **KTC_OPCODE_GT**
- const int **KTC_OPCODE_LE**
- const int **KTC_OPCODE_GE**
- const int **KTC_OPCODE_ASR**
- const int **KTC_OPCODE_ASL**
- const int **KTC_OPCODE_ADD**
- const int **KTC_OPCODE_SUB**
- const int **KTC_OPCODE_MUL**
- const int **KTC_OPCODE_DIV**
- const int **KTC_OPCODE_MOD**
- const int **KTC_OPCODE_PREINC**
- const int **KTC_OPCODE_PREDEC**
- const int **KTC_OPCODE_SIZEOF**
- const int **KTC_OPCODE_DEREF**
- const int **KTC_OPCODE_ADDRESS**
- const int **KTC_OPCODE_PLUS**
- const int **KTC_OPCODE_MINUS**
- const int **KTC_OPCODE_BITNOT**
- const int **KTC_OPCODE_LOGNOT**
- const int **KTC_OPCODE_POSTINC**
- const int **KTC_OPCODE_POSTDEC**
- const int **KTC_OPCODE_FIELD**
- const int **KTC_OPCODE_FIELDREF**

- const int **KTC_OPCODE_DOTAST**
- const int **KTC_OPCODE_DEREFAST**
- const int **KTC_OPCODE_ROUND_BRACKETS**
- const int **KTC_OPCODE_SQUARE_BRACKETS**
- const int **KTC_OPCODE_THROW**
- const int **KTC_OPCODE_MIN**
- const int **KTC_OPCODE_MAX**

4.16.1 Detailed Description

4.16.2 Variable Documentation

4.16.2.1 KTC_OPCODE_ADD

```
const int KTC_OPCODE_ADD
```

4.16.2.2 KTC_OPCODE_ADDASSIGN

```
const int KTC_OPCODE_ADDASSIGN
```

4.16.2.3 KTC_OPCODE_ADDRESS

```
const int KTC_OPCODE_ADDRESS
```

4.16.2.4 KTC_OPCODE_ANDASSIGN

```
const int KTC_OPCODE_ANDASSIGN
```

4.16.2.5 KTC_OPCODE_ASL

```
const int KTC_OPCODE_ASL
```

4.16.2.6 KTC_OPCODE_ASLASSIGN

```
const int KTC_OPCODE_ASLASSIGN
```

4.16.2.7 KTC_OPCODE_ASR

```
const int KTC_OPCODE_ASR
```

4.16.2.8 KTC_OPCODE_ASRASSIGN

```
const int KTC_OPCODE_ASRASSIGN
```

4.16.2.9 KTC_OPCODE_ASSIGN

```
const int KTC_OPCODE_ASSIGN
```

4.16.2.10 KTC_OPCODE_BITAND

```
const int KTC_OPCODE_BITAND
```

4.16.2.11 KTC_OPCODE_BITNOT

```
const int KTC_OPCODE_BITNOT
```

4.16.2.12 KTC_OPCODE_BITOR

```
const int KTC_OPCODE_BITOR
```

4.16.2.13 KTC_OPCODE_BITXOR

```
const int KTC_OPCODE_BITXOR
```

4.16.2.14 KTC_OPCODE_COMMA

```
const int KTC_OPCODE_COMMA
```

4.16.2.15 KTC_OPCODE_COND

```
const int KTC_OPCODE_COND
```

4.16.2.16 KTC_OPCODE_DEREF

```
const int KTC_OPCODE_DEREF
```

4.16.2.17 KTC_OPCODE_DEREFAST

```
const int KTC_OPCODE_DEREFAST
```

4.16.2.18 KTC_OPCODE_DIV

```
const int KTC_OPCODE_DIV
```

4.16.2.19 KTC_OPCODE_DIVASSIGN

```
const int KTC_OPCODE_DIVASSIGN
```

4.16.2.20 KTC_OPCODE_DOTAST

```
const int KTC_OPCODE_DOTAST
```

4.16.2.21 KTC_OPCODE_EQ

```
const int KTC_OPCODE_EQ
```

4.16.2.22 KTC_OPCODE_FIELD

```
const int KTC_OPCODE_FIELD
```

4.16.2.23 KTC_OPCODE_FIELDREF

```
const int KTC_OPCODE_FIELDREF
```

4.16.2.24 KTC_OPCODE_GE

```
const int KTC_OPCODE_GE
```

4.16.2.25 KTC_OPCODE_GT

```
const int KTC_OPCODE_GT
```

4.16.2.26 KTC_OPCODE_LE

```
const int KTC_OPCODE_LE
```

4.16.2.27 KTC_OPCODE_LOGAND

```
const int KTC_OPCODE_LOGAND
```

4.16.2.28 KTC_OPCODE_LOGNOT

```
const int KTC_OPCODE_LOGNOT
```

4.16.2.29 KTC_OPCODE_LOGOR

```
const int KTC_OPCODE_LOGOR
```

4.16.2.30 KTC_OPCODE_LT

```
const int KTC_OPCODE_LT
```

4.16.2.31 KTC_OPCODE_MAX

```
const int KTC_OPCODE_MAX
```

4.16.2.32 KTC_OPCODE_MIN

```
const int KTC_OPCODE_MIN
```

4.16.2.33 KTC_OPCODE_MINUS

```
const int KTC_OPCODE_MINUS
```

4.16.2.34 KTC_OPCODE_MOD

```
const int KTC_OPCODE_MOD
```

4.16.2.35 KTC_OPCODE_MODASSIGN

```
const int KTC_OPCODE_MODASSIGN
```

4.16.2.36 KTC_OPCODE_MUL

```
const int KTC_OPCODE_MUL
```

4.16.2.37 KTC_OPCODE_MULASSIGN

```
const int KTC_OPCODE_MULASSIGN
```

4.16.2.38 KTC_OPCODE_NE

```
const int KTC_OPCODE_NE
```

4.16.2.39 KTC_OPCODE_NONE

```
const int KTC_OPCODE_NONE
```

4.16.2.40 KTC_OPCODE_ORASSIGN

```
const int KTC_OPCODE_ORASSIGN
```

4.16.2.41 KTC_OPCODE_PLUS

```
const int KTC_OPCODE_PLUS
```

4.16.2.42 KTC_OPCODE_POSTDEC

```
const int KTC_OPCODE_POSTDEC
```

4.16.2.43 KTC_OPCODE_POSTINC

```
const int KTC_OPCODE_POSTINC
```

4.16.2.44 KTC_OPCODE_PREDEC

```
const int KTC_OPCODE_PREDEC
```

4.16.2.45 KTC_OPCODE_PREINC

```
const int KTC_OPCODE_PREINC
```

4.16.2.46 KTC_OPCODE_ROUND_BRACKETS

```
const int KTC_OPCODE_ROUND_BRACKETS
```

4.16.2.47 KTC_OPCODE_SIZEOF

```
const int KTC_OPCODE_SIZEOF
```

4.16.2.48 KTC_OPCODE_SQUARE_BRACKETS

```
const int KTC_OPCODE_SQUARE_BRACKETS
```

4.16.2.49 KTC_OPCODE_SUB

```
const int KTC_OPCODE_SUB
```

4.16.2.50 KTC_OPCODE_SUBASSIGN

```
const int KTC_OPCODE_SUBASSIGN
```

4.16.2.51 KTC_OPCODE_THROW

```
const int KTC_OPCODE_THROW
```

4.16.2.52 KTC_OPCODE_XORASSIGN

```
const int KTC_OPCODE_XORASSIGN
```

4.17 Numerical codes of declaration storage class specifiers

Modules

- **Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values**

Variables

- const int **KTC_STORAGECLASS_NONE**
- const int **KTC_STORAGECLASS_TYPEDEF**
- const int **KTC_STORAGECLASS_EXTERN**
- const int **KTC_STORAGECLASS_STATIC**
- const int **KTC_STORAGECLASS_AUTO**
- const int **KTC_STORAGECLASS_REGISTER**
- const int **KTC_STORAGECLASS_MUTABLE**
- const int **KTC_STORAGECLASS_THREADLOCAL**

4.17.1 Detailed Description

4.17.2 Variable Documentation

4.17.2.1 KTC_STORAGECLASS_AUTO

```
const int KTC_STORAGECLASS_AUTO
```

4.17.2.2 KTC_STORAGECLASS_EXTERN

```
const int KTC_STORAGECLASS_EXTERN
```

4.17.2.3 KTC_STORAGECLASS_MUTABLE

```
const int KTC_STORAGECLASS_MUTABLE
```

4.17.2.4 KTC_STORAGECLASS_NONE

```
const int KTC_STORAGECLASS_NONE
```

4.17.2.5 KTC_STORAGECLASS_REGISTER

```
const int KTC_STORAGECLASS_REGISTER
```

4.17.2.6 KTC_STORAGECLASS_STATIC

```
const int KTC_STORAGECLASS_STATIC
```

4.17.2.7 KTC_STORAGECLASS_THREADLOCAL

```
const int KTC_STORAGECLASS_THREADLOCAL
```

4.17.2.8 KTC_STORAGECLASS_TYPEDEF

```
const int KTC_STORAGECLASS_TYPEDEF
```

4.18 Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values

Modules

- **Numerical codes to differentiate struct/class/union declarations**

Variables

- const int **KTC_CVQUALIFIER_NONE**
- const int **KTC_CVQUALIFIER_CONST**
- const int **KTC_CVQUALIFIER_VOLATILE**
- const int **KTC_CVQUALIFIER_RESTRICT**

4.18.1 Detailed Description

4.18.2 Variable Documentation

4.18.2.1 KTC_CVQUALIFIER_CONST

```
const int KTC_CVQUALIFIER_CONST
```

4.18.2.2 KTC_CVQUALIFIER_NONE

```
const int KTC_CVQUALIFIER_NONE
```

4.18.2.3 KTC_CVQUALIFIER_RESTRICT

```
const int KTC_CVQUALIFIER_RESTRICT
```

4.18.2.4 KTC_CVQUALIFIER_VOLATILE

```
const int KTC_CVQUALIFIER_VOLATILE
```

4.19 Numerical codes to differentiate struct/class/union declarations

Modules

- **Numerical codes of C/C++ built-in types**

Variables

- const int **KTC_CLASSTAG_NONE**
- const int **KTC_CLASSTAG_STRUCT**
- const int **KTC_CLASSTAG_UNION**
- const int **KTC_CLASSTAG_CLASS**

4.19.1 Detailed Description

4.19.2 Variable Documentation

4.19.2.1 KTC_CLASSTAG_CLASS

```
const int KTC_CLASSTAG_CLASS
```

4.19.2.2 KTC_CLASSTAG_NONE

```
const int KTC_CLASSTAG_NONE
```

4.19.2.3 KTC_CLASSTAG_STRUCT

```
const int KTC_CLASSTAG_STRUCT
```

4.19.2.4 KTC_CLASSTAG_UNION

```
const int KTC_CLASSTAG_UNION
```

4.20 Numerical codes of C/C++ built-in types

Modules

- **Numerical codes for identifying inline/virtual/explicit/friend member function specifiers**

Variables

- const int **KTC_BUILTINTYPE_NONE**
- const int **KTC_BUILTINTYPE_VOID**
- const int **KTC_BUILTINTYPE_BOOL**
- const int **KTC_BUILTINTYPE_WCHAR_T**
- const int **KTC_BUILTINTYPE_CHAR**
- const int **KTC_BUILTINTYPE_SIGNEDCHAR**
- const int **KTC_BUILTINTYPE_UNSIGNEDCHAR**
- const int **KTC_BUILTINTYPE_SHORTINT**
- const int **KTC_BUILTINTYPE_SIGNEDSHORTINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDSHORTINT**
- const int **KTC_BUILTINTYPE_INT**
- const int **KTC_BUILTINTYPE_SIGNEDINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDINT**
- const int **KTC_BUILTINTYPE_LONGINT**
- const int **KTC_BUILTINTYPE_SIGNEDLONGINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDLONGINT**
- const int **KTC_BUILTINTYPE_LONGLONGINT**
- const int **KTC_BUILTINTYPE_SIGNEDLONGLONGINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDLONGLONGINT**
- const int **KTC_BUILTINTYPE_FLOAT**
- const int **KTC_BUILTINTYPE_DOUBLE**
- const int **KTC_BUILTINTYPE_LONGDOUBLE**
- const int **KTC_BUILTINTYPE_NULLPTR_T**

4.20.1 Detailed Description

4.20.2 Variable Documentation

4.20.2.1 KTC_BUILTINTYPE_BOOL

```
const int KTC_BUILTINTYPE_BOOL
```

4.20.2.2 KTC_BUILTINTYPE_CHAR

```
const int KTC_BUILTINTYPE_CHAR
```

4.20.2.3 KTC_BUILTINTYPE_DOUBLE

```
const int KTC_BUILTINTYPE_DOUBLE
```

4.20.2.4 KTC_BUILTINTYPE_FLOAT

```
const int KTC_BUILTINTYPE_FLOAT
```

4.20.2.5 KTC_BUILTINTYPE_INT

```
const int KTC_BUILTINTYPE_INT
```

4.20.2.6 KTC_BUILTINTYPE_LONGDOUBLE

```
const int KTC_BUILTINTYPE_LONGDOUBLE
```

4.20.2.7 KTC_BUILTINTYPE_LONGINT

```
const int KTC_BUILTINTYPE_LONGINT
```

4.20.2.8 KTC_BUILTINTYPE_LONGLONGINT

```
const int KTC_BUILTINTYPE_LONGLONGINT
```

4.20.2.9 KTC_BUILTINTYPE_NONE

```
const int KTC_BUILTINTYPE_NONE
```

4.20.2.10 KTC_BUILTINTYPE_NULLPTR_T

```
const int KTC_BUILTINTYPE_NULLPTR_T
```

4.20.2.11 KTC_BUILTINTYPE_SHORTINT

```
const int KTC_BUILTINTYPE_SHORTINT
```

4.20.2.12 KTC_BUILTINTYPE_SIGNEDCHAR

```
const int KTC_BUILTINTYPE_SIGNEDCHAR
```

4.20.2.13 KTC_BUILTINTYPE_SIGNEDINT

```
const int KTC_BUILTINTYPE_SIGNEDINT
```

4.20.2.14 KTC_BUILTINTYPE_SIGNEDLONGINT

```
const int KTC_BUILTINTYPE_SIGNEDLONGINT
```

4.20.2.15 KTC_BUILTINTYPE_SIGNEDLONGLONGINT

```
const int KTC_BUILTINTYPE_SIGNEDLONGLONGINT
```

4.20.2.16 KTC_BUILTINTYPE_SIGNEDSHORTINT

```
const int KTC_BUILTINTYPE_SIGNEDSHORTINT
```

4.20.2.17 KTC_BUILTINTYPE_UNSIGNEDCHAR

```
const int KTC_BUILTINTYPE_UNSIGNEDCHAR
```

4.20.2.18 KTC_BUILTINTYPE_UNSIGNEDINT

```
const int KTC_BUILTINTYPE_UNSIGNEDINT
```

4.20.2.19 KTC_BUILTINTYPE_UNSIGNEDLONGINT

```
const int KTC_BUILTINTYPE_UNSIGNEDLONGINT
```

4.20.2.20 KTC_BUILTINTYPE_UNSIGNEDLONGLONGINT

```
const int KTC_BUILTINTYPE_UNSIGNEDLONGLONGINT
```

4.20.2.21 KTC_BUILTINTYPE_UNSIGNEDSHORTINT

```
const int KTC_BUILTINTYPE_UNSIGNEDSHORTINT
```

4.20.2.22 KTC_BUILTINTYPE_VOID

```
const int KTC_BUILTINTYPE_VOID
```

4.20.2.23 KTC_BUILTINTYPE_WCHAR_T

```
const int KTC_BUILTINTYPE_WCHAR_T
```

4.21 Numerical codes for identifying inline/virtual/explicit/friend member function specifiers

Modules

- **Numerical codes for identifying different C++ -style cast expressions**

Variables

- const int **KTC_FUNCSPECIFIER_NONE**
- const int **KTC_FUNCSPECIFIER_INLINE**
- const int **KTC_FUNCSPECIFIER_VIRTUAL**
- const int **KTC_FUNCSPECIFIER_EXPLICIT**
- const int **KTC_FUNCSPECIFIER_FRIEND**

4.21.1 Detailed Description

4.21.2 Variable Documentation

4.21.2.1 KTC_FUNCSPECIFIER_EXPLICIT

```
const int KTC_FUNCSPECIFIER_EXPLICIT
```

4.21.2.2 KTC_FUNCSPECIFIER_FRIEND

```
const int KTC_FUNCSPECIFIER_FRIEND
```

4.21.2.3 KTC_FUNCSPECIFIER_INLINE

```
const int KTC_FUNCSPECIFIER_INLINE
```

4.21.2.4 KTC_FUNCSPECIFIER_NONE

```
const int KTC_FUNCSPECIFIER_NONE
```

4.21.2.5 KTC_FUNCSPECIFIER_VIRTUAL

```
const int KTC_FUNCSPECIFIER_VIRTUAL
```

4.22 Numerical codes for identifying different C++ -style cast expressions

Modules

- **Numerical codes for identifying declarator ptr types**

Variables

- const int **KTC_CASTSPECIFIER_DYNAMIC**
- const int **KTC_CASTSPECIFIER_STATIC**
- const int **KTC_CASTSPECIFIER_REINTERPRET**
- const int **KTC_CASTSPECIFIER_CONST**

4.22.1 Detailed Description

4.22.2 Variable Documentation

4.22.2.1 KTC_CASTSPECIFIER_CONST

```
const int KTC_CASTSPECIFIER_CONST
```

4.22.2.2 KTC_CASTSPECIFIER_DYNAMIC

```
const int KTC_CASTSPECIFIER_DYNAMIC
```

4.22.2.3 KTC_CASTSPECIFIER_REINTERPRET

```
const int KTC_CASTSPECIFIER_REINTERPRET
```

4.22.2.4 KTC_CASTSPECIFIER_STATIC

```
const int KTC_CASTSPECIFIER_STATIC
```

4.23 Numerical codes for identifying declarator ptr types

Functions

- intptr_t **ktc_get_child_index** (**ktc_treeType_t** tt, **ktc_childId_t** child_id)

Variables

- const int **KTC_POINTEROPERATOR_NONE**
- const int **KTC_POINTEROPERATOR_POINTER**
- const int **KTC_POINTEROPERATOR_REFERENCE**
- const int **KTC_POINTEROPERATOR_RVALUE**

4.23.1 Detailed Description

4.23.2 Function Documentation

4.23.2.1 ktc_get_child_index()

```
intptr_t ktc_get_child_index (
            ktc_treeType_t tt,
            ktc_childId_t child_id )
```

4.23.3 Variable Documentation

4.23.3.1 KTC_POINTEROPERATOR_NONE

```
const int KTC_POINTEROPERATOR_NONE
```

4.23.3.2 KTC_POINTEROPERATOR_POINTER

```
const int KTC_POINTEROPERATOR_POINTER
```

4.23.3.3 KTC_POINTEROPERATOR_REFERENCE

```
const int KTC_POINTEROPERATOR_REFERENCE
```

4.23.3.4 KTC_POINTEROPERATOR_RVALUE

```
const int KTC_POINTEROPERATOR_RVALUE
```

Chapter 5

File Documentation

5.1 gen-ktcAPI.h File Reference

```
#include "kwapi.h"
#include "ktcMainAPI.h"
#include "stddef.h"
```

Functions

- int **ktc_is_Node** (**ktc_tree_t** t)
- int **ktc_is_TranslationUnit** (**ktc_tree_t** t)
- int **ktc_is_DeclOrStmts** (**ktc_tree_t** t)
- int **ktc_is_AnyFuncBody** (**ktc_tree_t** t)
- int **ktc_is_Handlers** (**ktc_tree_t** t)
- int **ktc_is_ExceptHandler** (**ktc_tree_t** t)
- int **ktc_is_FinallyHandler** (**ktc_tree_t** t)
- int **ktc_is_MaybeException** (**ktc_tree_t** t)
- int **ktc_is_TemplateParams** (**ktc_tree_t** t)
- int **ktc_is_DeclSpecs** (**ktc_tree_t** t)
- int **ktc_is_AttributeSpecs** (**ktc_tree_t** t)
- int **ktc_is_Attributes** (**ktc_tree_t** t)
- int **ktc_is_PropertyFuncs** (**ktc_tree_t** t)
- int **ktc_is_MemberDecls** (**ktc_tree_t** t)
- int **ktc_is_Enumerators** (**ktc_tree_t** t)
- int **ktc_is_MaybeDeclarator** (**ktc_tree_t** t)
- int **ktc_is_ParamNames** (**ktc_tree_t** t)
- int **ktc_is_Initializers** (**ktc_tree_t** t)
- int **ktc_is_Designators** (**ktc_tree_t** t)
- int **ktc_is_AnyLabel** (**ktc_tree_t** t)
- int **ktc_is_Exprs** (**ktc_tree_t** t)
- int **ktc_is_AnyNameQualifier** (**ktc_tree_t** t)
- int **ktc_is_AnyNames** (**ktc_tree_t** t)
- int **ktc_is_TemplateArgs** (**ktc_tree_t** t)
- int **ktc_is_BaseSpecs** (**ktc_tree_t** t)
- int **ktc_is_MaybeLambdaDeclarator** (**ktc_tree_t** t)
- int **ktc_is_LambdaIntroducer** (**ktc_tree_t** t)

- int **ktc_is_AnyCapture** (**ktc_tree_t** t)
- int **ktc_is_MaybeTypeId** (**ktc_tree_t** t)
- int **ktc_is_MaybeNewInitializer** (**ktc_tree_t** t)
- int **ktc_is_MemberInitializers** (**ktc_tree_t** t)
- int **ktc_is_MaybeCtorInitializer** (**ktc_tree_t** t)
- int **ktc_is_MaybeExceptionSpec** (**ktc_tree_t** t)
- int **ktc_is_NoDeclOrStmt** (**ktc_tree_t** t)
- int **ktc_is_DeclEllipsis** (**ktc_tree_t** t)
- int **ktc_is_DeclOrStmt** (**ktc_tree_t** t)
- int **ktc_is_FuncBody** (**ktc_tree_t** t)
- int **ktc_is_FuncTryBlock** (**ktc_tree_t** t)
- int **ktc_is_PromisedFuncBody** (**ktc_tree_t** t)
- int **ktc_is_NoHandler** (**ktc_tree_t** t)
- int **ktc_is_Handler** (**ktc_tree_t** t)
- int **ktc_is_NoException** (**ktc_tree_t** t)
- int **ktc_is_Exception** (**ktc_tree_t** t)
- int **ktc_is_DefaultException** (**ktc_tree_t** t)
- int **ktc_is_UnparsedException** (**ktc_tree_t** t)
- int **ktc_is_NoTemplateParam** (**ktc_tree_t** t)
- int **ktc_is_TemplateParam** (**ktc_tree_t** t)
- int **ktc_is_NoDeclSpec** (**ktc_tree_t** t)
- int **ktc_is_DeclSpec** (**ktc_tree_t** t)
- int **ktc_is_NoAttributeSpec** (**ktc_tree_t** t)
- int **ktc_is_AttributeSpec** (**ktc_tree_t** t)
- int **ktc_is_NoAttribute** (**ktc_tree_t** t)
- int **ktc_is_AnyAttribute** (**ktc_tree_t** t)
- int **ktc_is_NoPropertyFunc** (**ktc_tree_t** t)
- int **ktc_is_AnyPropertyFunc** (**ktc_tree_t** t)
- int **ktc_is_NoMemberDecl** (**ktc_tree_t** t)
- int **ktc_is_AnyMemberDecl** (**ktc_tree_t** t)
- int **ktc_is_NoEnumerator** (**ktc_tree_t** t)
- int **ktc_is_AnyEnumerator** (**ktc_tree_t** t)
- int **ktc_is_NoDeclarator** (**ktc_tree_t** t)
- int **ktc_is_AnyDeclarator** (**ktc_tree_t** t)
- int **ktc_is_NoParamName** (**ktc_tree_t** t)
- int **ktc_is_AnyParamName** (**ktc_tree_t** t)
- int **ktc_is_NoInitializer** (**ktc_tree_t** t)
- int **ktc_is_AnyInitializer** (**ktc_tree_t** t)
- int **ktc_is_NoDesignator** (**ktc_tree_t** t)
- int **ktc_is_AnyDesignator** (**ktc_tree_t** t)
- int **ktc_is_Label** (**ktc_tree_t** t)
- int **ktc_is_CaseLabel** (**ktc_tree_t** t)
- int **ktc_is_DefaultLabel** (**ktc_tree_t** t)
- int **ktc_is_CaseRangeLabel** (**ktc_tree_t** t)
- int **ktc_is_UnparsedLabel** (**ktc_tree_t** t)
- int **ktc_is_NoExpr** (**ktc_tree_t** t)
- int **ktc_is_AnyExpr** (**ktc_tree_t** t)
- int **ktc_is_NoNameQualifier** (**ktc_tree_t** t)
- int **ktc_is_GlobalScope** (**ktc_tree_t** t)
- int **ktc_is_SuperScope** (**ktc_tree_t** t)
- int **ktc_is_AnyNameSpec** (**ktc_tree_t** t)
- int **ktc_is_NoName** (**ktc_tree_t** t)
- int **ktc_is_AnyName** (**ktc_tree_t** t)
- int **ktc_is_NoTemplateArg** (**ktc_tree_t** t)
- int **ktc_is_AnyTemplateArg** (**ktc_tree_t** t)

- int **ktc_is_NoBaseSpec** (**ktc_tree_t** t)
- int **ktc_is_BaseSpec** (**ktc_tree_t** t)
- int **ktc_is_NoLambdaDeclarator** (**ktc_tree_t** t)
- int **ktc_is_LambdaDeclarator** (**ktc_tree_t** t)
- int **ktc_is_CaptureDefault** (**ktc_tree_t** t)
- int **ktc_is_Capture** (**ktc_tree_t** t)
- int **ktc_is_NoCapture** (**ktc_tree_t** t)
- int **ktc_is_NoTypeId** (**ktc_tree_t** t)
- int **ktc_is_TypeId** (**ktc_tree_t** t)
- int **ktc_is_NoNewInitializer** (**ktc_tree_t** t)
- int **ktc_is_NewInitializer** (**ktc_tree_t** t)
- int **ktc_is_NoMemberInitializer** (**ktc_tree_t** t)
- int **ktc_is_MemberInitializer** (**ktc_tree_t** t)
- int **ktc_is_NoCtorInitializer** (**ktc_tree_t** t)
- int **ktc_is_CtorInitializer** (**ktc_tree_t** t)
- int **ktc_is_NoExceptionSpec** (**ktc_tree_t** t)
- int **ktc_is_ExceptionSpec** (**ktc_tree_t** t)
- int **ktc_is_DenyThrowSpec** (**ktc_tree_t** t)
- int **ktc_is_AnyDecl** (**ktc_tree_t** t)
- int **ktc_is_AnyStmt** (**ktc_tree_t** t)
- int **ktc_is_AnyTypeParam** (**ktc_tree_t** t)
- int **ktc_is_Param** (**ktc_tree_t** t)
- int **ktc_is_AutoType** (**ktc_tree_t** t)
- int **ktc_is_ConstExpr** (**ktc_tree_t** t)
- int **ktc_is_StorageClass** (**ktc_tree_t** t)
- int **ktc_is_AlignAsExpr** (**ktc_tree_t** t)
- int **ktc_is_AlignAsType** (**ktc_tree_t** t)
- int **ktc_is_AnyTypeName** (**ktc_tree_t** t)
- int **ktc_is_FuncSpec** (**ktc_tree_t** t)
- int **ktc_is_AttributeDeclSpec** (**ktc_tree_t** t)
- int **ktc_is_UnparsedDeclSpec** (**ktc_tree_t** t)
- int **ktc_is_Attribute** (**ktc_tree_t** t)
- int **ktc_is_AttributeWithArgs** (**ktc_tree_t** t)
- int **ktc_is_PropertyAttribute** (**ktc_tree_t** t)
- int **ktc_is_GenericAttribute** (**ktc_tree_t** t)
- int **ktc_is_PropertyPutFunc** (**ktc_tree_t** t)
- int **ktc_is_PropertyGetFunc** (**ktc_tree_t** t)
- int **ktc_is_UnparsedPropertyFunc** (**ktc_tree_t** t)
- int **ktc_is_MemberDecl** (**ktc_tree_t** t)
- int **ktc_is_MemberFunc** (**ktc_tree_t** t)
- int **ktc_is_AccessSpecification** (**ktc_tree_t** t)
- int **ktc_is_MemberTemplate** (**ktc_tree_t** t)
- int **ktc_is_MemberUsingDecl** (**ktc_tree_t** t)
- int **ktc_is_UnparsedMemberDecl** (**ktc_tree_t** t)
- int **ktc_is_PromisedMemberDecl** (**ktc_tree_t** t)
- int **ktc_is_Enumerator** (**ktc_tree_t** t)
- int **ktc_is_UnparsedEnumerator** (**ktc_tree_t** t)
- int **ktc_is_AnyNonPtrDeclarator** (**ktc_tree_t** t)
- int **ktc_is_PtrDeclarator** (**ktc_tree_t** t)
- int **ktc_is_BitFieldDeclarator** (**ktc_tree_t** t)
- int **ktc_is_AttributedDeclarator** (**ktc_tree_t** t)
- int **ktc_is_InitializedDeclarator** (**ktc_tree_t** t)
- int **ktc_is_UnparsedDeclarator** (**ktc_tree_t** t)
- int **ktc_is_ParamName** (**ktc_tree_t** t)
- int **ktc_is_UnparsedParamName** (**ktc_tree_t** t)

- int **ktc_is_CopyInitializer** (**ktc_tree_t** t)
- int **ktc_is_InitClause** (**ktc_tree_t** t)
- int **ktc_is_TruncatedInitClause** (**ktc_tree_t** t)
- int **ktc_is_DirectInitializer** (**ktc_tree_t** t)
- int **ktc_is_UnparsedInitializer** (**ktc_tree_t** t)
- int **ktc_is_FieldDesignator** (**ktc_tree_t** t)
- int **ktc_is_MemberDesignator** (**ktc_tree_t** t)
- int **ktc_is_IndexDesignator** (**ktc_tree_t** t)
- int **ktc_is_RangeDesignator** (**ktc_tree_t** t)
- int **ktc_is_IdExpr** (**ktc_tree_t** t)
- int **ktc_is_BoolLiteralExpr** (**ktc_tree_t** t)
- int **ktc_is_LiteralExpr** (**ktc_tree_t** t)
- int **ktc_is_UserLiteralExpr** (**ktc_tree_t** t)
- int **ktc_is_StringLiteralExpr** (**ktc_tree_t** t)
- int **ktc_is_UserStringLiteralExpr** (**ktc_tree_t** t)
- int **ktc_is_NullptrLiteralExpr** (**ktc_tree_t** t)
- int **ktc_is_MemberExpr** (**ktc_tree_t** t)
- int **ktc_is_CallExpr** (**ktc_tree_t** t)
- int **ktc_is_TypeConvExpr** (**ktc_tree_t** t)
- int **ktc_is_IndexExpr** (**ktc_tree_t** t)
- int **ktc_is_UnaryExpr** (**ktc_tree_t** t)
- int **ktc_is_ThrowExpr** (**ktc_tree_t** t)
- int **ktc_is_BinaryExpr** (**ktc_tree_t** t)
- int **ktc_is_ConditionalExpr** (**ktc_tree_t** t)
- int **ktc_is_SizeOfExpr** (**ktc_tree_t** t)
- int **ktc_is_AlignOfExpr** (**ktc_tree_t** t)
- int **ktc_is_CastExpr** (**ktc_tree_t** t)
- int **ktc_is_SpecialCastExpr** (**ktc_tree_t** t)
- int **ktc_is_ParensExpr** (**ktc_tree_t** t)
- int **ktc_is_ThisExpr** (**ktc_tree_t** t)
- int **ktc_is_NewExpr** (**ktc_tree_t** t)
- int **ktc_is_DeleteExpr** (**ktc_tree_t** t)
- int **ktc_is_TypeTypeIdExpr** (**ktc_tree_t** t)
- int **ktc_is_ExprTypeIdExpr** (**ktc_tree_t** t)
- int **ktc_is_StmtExpr** (**ktc_tree_t** t)
- int **ktc_is_InitializerExpr** (**ktc_tree_t** t)
- int **ktc_is_LambdaExpr** (**ktc_tree_t** t)
- int **ktc_is_UnparsedExpr** (**ktc_tree_t** t)
- int **ktc_is_NameSpec** (**ktc_tree_t** t)
- int **ktc_is_TemplateSpec** (**ktc_tree_t** t)
- int **ktc_is_TypeOfSpec** (**ktc_tree_t** t)
- int **ktc_is_UnparsedNameQualifier** (**ktc_tree_t** t)
- int **ktc_is_QualifiedName** (**ktc_tree_t** t)
- int **ktc_is_UnqualifiedName** (**ktc_tree_t** t)
- int **ktc_is_ExprArg** (**ktc_tree_t** t)
- int **ktc_is_TypeArg** (**ktc_tree_t** t)
- int **ktc_is_TemplateTypeArg** (**ktc_tree_t** t)
- int **ktc_is_FuncDef** (**ktc_tree_t** t)
- int **ktc_is_Decl** (**ktc_tree_t** t)
- int **ktc_is_TemplateDecl** (**ktc_tree_t** t)
- int **ktc_is_LinkageSpec** (**ktc_tree_t** t)
- int **ktc_is_ExplicitInstantiation** (**ktc_tree_t** t)
- int **ktc_is_NamespaceDecl** (**ktc_tree_t** t)
- int **ktc_is_NamespaceAlias** (**ktc_tree_t** t)
- int **ktc_is_AnyUsing** (**ktc_tree_t** t)

- int **ktc_is_AliasDecl** (**ktc_tree_t** t)
- int **ktc_is_AsmDef** (**ktc_tree_t** t)
- int **ktc_is_StaticAssertDecl** (**ktc_tree_t** t)
- int **ktc_is_UnparsedDecl** (**ktc_tree_t** t)
- int **ktc_is_LabeledStmt** (**ktc_tree_t** t)
- int **ktc_is_ExprStmt** (**ktc_tree_t** t)
- int **ktc_is_CompoundStmt** (**ktc_tree_t** t)
- int **ktc_is_IfStmt** (**ktc_tree_t** t)
- int **ktc_is_IfDeclStmt** (**ktc_tree_t** t)
- int **ktc_is_SwitchStmt** (**ktc_tree_t** t)
- int **ktc_is_SwitchDeclStmt** (**ktc_tree_t** t)
- int **ktc_is_WhileStmt** (**ktc_tree_t** t)
- int **ktc_is_WhileDeclStmt** (**ktc_tree_t** t)
- int **ktc_is_DoStmt** (**ktc_tree_t** t)
- int **ktc_is_DoDeclStmt** (**ktc_tree_t** t)
- int **ktc_is_ForStmt** (**ktc_tree_t** t)
- int **ktc_is_ForEachStmt** (**ktc_tree_t** t)
- int **ktc_is_ForRangeStmt** (**ktc_tree_t** t)
- int **ktc_is_GotoStmt** (**ktc_tree_t** t)
- int **ktc_is_ContinueStmt** (**ktc_tree_t** t)
- int **ktc_is_BreakStmt** (**ktc_tree_t** t)
- int **ktc_is_ReturnStmt** (**ktc_tree_t** t)
- int **ktc_is_TryStmt** (**ktc_tree_t** t)
- int **ktc_is_TryExceptStmt** (**ktc_tree_t** t)
- int **ktc_is_TryFinallyStmt** (**ktc_tree_t** t)
- int **ktc_is_LeaveStmt** (**ktc_tree_t** t)
- int **ktc_is_AsmStmt** (**ktc_tree_t** t)
- int **ktc_is_UnparsedStmt** (**ktc_tree_t** t)
- int **ktc_is_TypeParam** (**ktc_tree_t** t)
- int **ktc_is_TemplateTypeParam** (**ktc_tree_t** t)
- int **ktc_is_CVQualifier** (**ktc_tree_t** t)
- int **ktc_is_ReservedTypeSpec** (**ktc_tree_t** t)
- int **ktc_is_TypeName** (**ktc_tree_t** t)
- int **ktc_is_EnumType** (**ktc_tree_t** t)
- int **ktc_is_ClassType** (**ktc_tree_t** t)
- int **ktc_is_AnyTypeOf** (**ktc_tree_t** t)
- int **ktc_is_NameDeclarator** (**ktc_tree_t** t)
- int **ktc_is_ParensDeclarator** (**ktc_tree_t** t)
- int **ktc_is_ArrayDeclarator** (**ktc_tree_t** t)
- int **ktc_is_FuncDeclarator** (**ktc_tree_t** t)
- int **ktc_is_KRFuncDeclarator** (**ktc_tree_t** t)
- int **ktc_is_Name** (**ktc_tree_t** t)
- int **ktc_is_Dtor** (**ktc_tree_t** t)
- int **ktc_is_SuffixFunc** (**ktc_tree_t** t)
- int **ktc_is_OpFunc** (**ktc_tree_t** t)
- int **ktc_is_ConvFunc** (**ktc_tree_t** t)
- int **ktc_is_TemplateName** (**ktc_tree_t** t)
- int **ktc_is_AnyPseudoDtor** (**ktc_tree_t** t)
- int **ktc_is_UnparsedName** (**ktc_tree_t** t)
- int **ktc_is_UsingDecl** (**ktc_tree_t** t)
- int **ktc_is_UsingDirective** (**ktc_tree_t** t)
- int **ktc_is_TypeAdjective** (**ktc_tree_t** t)
- int **ktc_is_BuiltinType** (**ktc_tree_t** t)
- int **ktc_is_TypeOfExpr** (**ktc_tree_t** t)
- int **ktc_is_TypeOfType** (**ktc_tree_t** t)
- int **ktc_is_PseudoDtor** (**ktc_tree_t** t)
- int **ktc_is_QualifiedPseudoDtor** (**ktc_tree_t** t)
- intptr_t **ktc_get_child_index** (**ktc_treeType_t** tt, **ktc_childId_t** child_id)

Variables

- const **ktc_treeType_t tid_Any**
- const **ktc_treeType_t tid_Node**
- const **ktc_treeType_t tid_TranslationUnit**
- const **ktc_treeType_t tid_DeclOrStmts**
- const **ktc_treeType_t tid_AnyFuncBody**
- const **ktc_treeType_t tid_Handlers**
- const **ktc_treeType_t tid_ExceptHandler**
- const **ktc_treeType_t tid_FinallyHandler**
- const **ktc_treeType_t tid_MaybeException**
- const **ktc_treeType_t tid_TemplateParams**
- const **ktc_treeType_t tid_DeclSpecs**
- const **ktc_treeType_t tid_AttributeSpecs**
- const **ktc_treeType_t tid_Attributes**
- const **ktc_treeType_t tid_PropertyFuncs**
- const **ktc_treeType_t tid_MemberDecls**
- const **ktc_treeType_t tid_Enumerators**
- const **ktc_treeType_t tid_MaybeDeclarator**
- const **ktc_treeType_t tid_ParamNames**
- const **ktc_treeType_t tid_Initializers**
- const **ktc_treeType_t tid_Designators**
- const **ktc_treeType_t tid_AnyLabel**
- const **ktc_treeType_t tid_Exprs**
- const **ktc_treeType_t tid_AnyNameQualifier**
- const **ktc_treeType_t tid_AnyNames**
- const **ktc_treeType_t tid_TemplateArgs**
- const **ktc_treeType_t tid_BaseSpecs**
- const **ktc_treeType_t tid_MaybeLambdaDeclarator**
- const **ktc_treeType_t tid_LambdaIntroducer**
- const **ktc_treeType_t tid_AnyCapture**
- const **ktc_treeType_t tid_MaybeTypeId**
- const **ktc_treeType_t tid_MaybeNewInitializer**
- const **ktc_treeType_t tid_MemberInitializers**
- const **ktc_treeType_t tid_MaybeCtorInitializer**
- const **ktc_treeType_t tid_MaybeExceptionSpec**
- const **ktc_treeType_t tid_NoDeclOrStmt**
- const **ktc_treeType_t tid_DeclEllipsis**
- const **ktc_treeType_t tid_DeclOrStmt**
- const **ktc_treeType_t tid_FuncBody**
- const **ktc_treeType_t tid_FuncTryBlock**
- const **ktc_treeType_t tid_PromisedFuncBody**
- const **ktc_treeType_t tid_NoHandler**
- const **ktc_treeType_t tid_Handler**
- const **ktc_treeType_t tid_NoException**
- const **ktc_treeType_t tid_Exception**
- const **ktc_treeType_t tid_DefaultException**
- const **ktc_treeType_t tid_UnparsedException**
- const **ktc_treeType_t tid_NoTemplateParam**
- const **ktc_treeType_t tid_TemplateParam**
- const **ktc_treeType_t tid_NoDeclSpec**
- const **ktc_treeType_t tid_DeclSpec**
- const **ktc_treeType_t tid_NoAttributeSpec**
- const **ktc_treeType_t tid_AttributeSpec**
- const **ktc_treeType_t tid_NoAttribute**

- const **ktc_treeType_t tid_AnyAttribute**
- const **ktc_treeType_t tid_NoPropertyFunc**
- const **ktc_treeType_t tid_AnyPropertyFunc**
- const **ktc_treeType_t tid_NoMemberDecl**
- const **ktc_treeType_t tid_AnyMemberDecl**
- const **ktc_treeType_t tid_NoEnumerator**
- const **ktc_treeType_t tid_AnyEnumerator**
- const **ktc_treeType_t tid_NoDeclarator**
- const **ktc_treeType_t tid_AnyDeclarator**
- const **ktc_treeType_t tid_NoParamName**
- const **ktc_treeType_t tid_AnyParamName**
- const **ktc_treeType_t tid_NoInitializer**
- const **ktc_treeType_t tid_AnyInitializer**
- const **ktc_treeType_t tid_NoDesignator**
- const **ktc_treeType_t tid_AnyDesignator**
- const **ktc_treeType_t tid_Label**
- const **ktc_treeType_t tid_CaseLabel**
- const **ktc_treeType_t tid_DefaultLabel**
- const **ktc_treeType_t tid_CaseRangeLabel**
- const **ktc_treeType_t tid_UnparsedLabel**
- const **ktc_treeType_t tid_NoExpr**
- const **ktc_treeType_t tid_AnyExpr**
- const **ktc_treeType_t tid_NoNameQualifier**
- const **ktc_treeType_t tid_GlobalScope**
- const **ktc_treeType_t tid_SuperScope**
- const **ktc_treeType_t tid_AnyNameSpec**
- const **ktc_treeType_t tid_NoName**
- const **ktc_treeType_t tid_AnyName**
- const **ktc_treeType_t tid_NoTemplateArg**
- const **ktc_treeType_t tid_AnyTemplateArg**
- const **ktc_treeType_t tid_NoBaseSpec**
- const **ktc_treeType_t tid_BaseSpec**
- const **ktc_treeType_t tid_NoLambdaDeclarator**
- const **ktc_treeType_t tid_LambdaDeclarator**
- const **ktc_treeType_t tid_CaptureDefault**
- const **ktc_treeType_t tid_Capture**
- const **ktc_treeType_t tid_NoCapture**
- const **ktc_treeType_t tid_NoTypeId**
- const **ktc_treeType_t tid_TypeId**
- const **ktc_treeType_t tid_NoNewInitializer**
- const **ktc_treeType_t tid_NewInitializer**
- const **ktc_treeType_t tid_NoMemberInitializer**
- const **ktc_treeType_t tid_MemberInitializer**
- const **ktc_treeType_t tid_NoCtorInitializer**
- const **ktc_treeType_t tid_CtorInitializer**
- const **ktc_treeType_t tid_NoExceptionSpec**
- const **ktc_treeType_t tid_ExceptionSpec**
- const **ktc_treeType_t tid_DenyThrowSpec**
- const **ktc_treeType_t tid_AnyDecl**
- const **ktc_treeType_t tid_AnyStmt**
- const **ktc_treeType_t tid_AnyTypeParam**
- const **ktc_treeType_t tid_Param**
- const **ktc_treeType_t tid_AutoType**
- const **ktc_treeType_t tid_ConstExpr**
- const **ktc_treeType_t tid_StorageClass**

- const **ktc_treeType_t tid_AlignAsExpr**
- const **ktc_treeType_t tid_AlignAsType**
- const **ktc_treeType_t tid_AnyTypeName**
- const **ktc_treeType_t tid_FuncSpec**
- const **ktc_treeType_t tid_AttributeDeclSpec**
- const **ktc_treeType_t tid_UnparsedDeclSpec**
- const **ktc_treeType_t tid_Attribute**
- const **ktc_treeType_t tid_AttributeWithArgs**
- const **ktc_treeType_t tid_PropertyAttribute**
- const **ktc_treeType_t tid_GenericAttribute**
- const **ktc_treeType_t tid_PropertyPutFunc**
- const **ktc_treeType_t tid_PropertyGetFunc**
- const **ktc_treeType_t tid_UnparsedPropertyFunc**
- const **ktc_treeType_t tid_MemberDecl**
- const **ktc_treeType_t tid_MemberFunc**
- const **ktc_treeType_t tid_AccessSpecification**
- const **ktc_treeType_t tid_MemberTemplate**
- const **ktc_treeType_t tid_MemberUsingDecl**
- const **ktc_treeType_t tid_UnparsedMemberDecl**
- const **ktc_treeType_t tid_PromisedMemberDecl**
- const **ktc_treeType_t tid_Enumerator**
- const **ktc_treeType_t tid_UnparsedEnumerator**
- const **ktc_treeType_t tid_AnyNonPtrDeclarator**
- const **ktc_treeType_t tid_PtrDeclarator**
- const **ktc_treeType_t tid_BitFieldDeclarator**
- const **ktc_treeType_t tid_AttributedDeclarator**
- const **ktc_treeType_t tid_InitializedDeclarator**
- const **ktc_treeType_t tid_UnparsedDeclarator**
- const **ktc_treeType_t tid_ParamName**
- const **ktc_treeType_t tid_UnparsedParamName**
- const **ktc_treeType_t tid_CopyInitializer**
- const **ktc_treeType_t tid_InitClause**
- const **ktc_treeType_t tid_TruncatedInitClause**
- const **ktc_treeType_t tid_DirectInitializer**
- const **ktc_treeType_t tid_UnparsedInitializer**
- const **ktc_treeType_t tid_FieldDesignator**
- const **ktc_treeType_t tid_MemberDesignator**
- const **ktc_treeType_t tid_IndexDesignator**
- const **ktc_treeType_t tid_RangeDesignator**
- const **ktc_treeType_t tid_IdExpr**
- const **ktc_treeType_t tid_BoolLiteralExpr**
- const **ktc_treeType_t tid_LiteralExpr**
- const **ktc_treeType_t tid_UserLiteralExpr**
- const **ktc_treeType_t tid_StringLiteralExpr**
- const **ktc_treeType_t tid_UserStringLiteralExpr**
- const **ktc_treeType_t tid_NullptrLiteralExpr**
- const **ktc_treeType_t tid_MemberExpr**
- const **ktc_treeType_t tid_CallExpr**
- const **ktc_treeType_t tid_TypeConvExpr**
- const **ktc_treeType_t tid_IndexExpr**
- const **ktc_treeType_t tid_UnaryExpr**
- const **ktc_treeType_t tid_ThrowExpr**
- const **ktc_treeType_t tid_BinaryExpr**
- const **ktc_treeType_t tid_ConditionalExpr**
- const **ktc_treeType_t tid_SizeOfExpr**

- const **ktc_treeType_t tid_AlignOfExpr**
- const **ktc_treeType_t tid_CastExpr**
- const **ktc_treeType_t tid_SpecialCastExpr**
- const **ktc_treeType_t tid_ParensExpr**
- const **ktc_treeType_t tid_ThisExpr**
- const **ktc_treeType_t tid_NewExpr**
- const **ktc_treeType_t tid_DeleteExpr**
- const **ktc_treeType_t tid_TypeTypeIdExpr**
- const **ktc_treeType_t tid_ExprTypeIdExpr**
- const **ktc_treeType_t tid_StmtExpr**
- const **ktc_treeType_t tid_InitializerExpr**
- const **ktc_treeType_t tid_LambdaExpr**
- const **ktc_treeType_t tid_UnparsedExpr**
- const **ktc_treeType_t tid_NameSpec**
- const **ktc_treeType_t tid_TemplateSpec**
- const **ktc_treeType_t tid_TypeOfSpec**
- const **ktc_treeType_t tid_UnparsedNameQualifier**
- const **ktc_treeType_t tid_QualifiedName**
- const **ktc_treeType_t tid_UnqualifiedName**
- const **ktc_treeType_t tid_ExprArg**
- const **ktc_treeType_t tid_TypeArg**
- const **ktc_treeType_t tid_TemplateTypeArg**
- const **ktc_treeType_t tid_FuncDef**
- const **ktc_treeType_t tid_Decl**
- const **ktc_treeType_t tid_TemplateDecl**
- const **ktc_treeType_t tid_LinkageSpec**
- const **ktc_treeType_t tid_ExplicitInstantiation**
- const **ktc_treeType_t tid_NamespaceDecl**
- const **ktc_treeType_t tid_NamespaceAlias**
- const **ktc_treeType_t tid_AnyUsing**
- const **ktc_treeType_t tid_AliasDecl**
- const **ktc_treeType_t tid_AsmDef**
- const **ktc_treeType_t tid_StaticAssertDecl**
- const **ktc_treeType_t tid_UnparsedDecl**
- const **ktc_treeType_t tid_LabeledStmt**
- const **ktc_treeType_t tid_ExprStmt**
- const **ktc_treeType_t tid_CompoundStmt**
- const **ktc_treeType_t tid_IfStmt**
- const **ktc_treeType_t tid_IfDeclStmt**
- const **ktc_treeType_t tid_SwitchStmt**
- const **ktc_treeType_t tid_SwitchDeclStmt**
- const **ktc_treeType_t tid_WhileStmt**
- const **ktc_treeType_t tid_WhileDeclStmt**
- const **ktc_treeType_t tid_DoStmt**
- const **ktc_treeType_t tid_DoDeclStmt**
- const **ktc_treeType_t tid_ForStmt**
- const **ktc_treeType_t tid_ForEachStmt**
- const **ktc_treeType_t tid_ForRangeStmt**
- const **ktc_treeType_t tid_GotoStmt**
- const **ktc_treeType_t tid_ContinueStmt**
- const **ktc_treeType_t tid_BreakStmt**
- const **ktc_treeType_t tid_ReturnStmt**
- const **ktc_treeType_t tid_TryStmt**
- const **ktc_treeType_t tid_TryExceptStmt**
- const **ktc_treeType_t tid_TryFinallyStmt**

- const **ktc_treeType_t tid_LeaveStmt**
- const **ktc_treeType_t tid_AsmStmt**
- const **ktc_treeType_t tid_UnparsedStmt**
- const **ktc_treeType_t tid_TypeParam**
- const **ktc_treeType_t tid_TemplateTypeParam**
- const **ktc_treeType_t tid_CVQualifier**
- const **ktc_treeType_t tid_ReservedTypeSpec**
- const **ktc_treeType_t tid_TypeName**
- const **ktc_treeType_t tid_EnumType**
- const **ktc_treeType_t tid_ClassType**
- const **ktc_treeType_t tid_AnyTypeOf**
- const **ktc_treeType_t tid_NameDeclarator**
- const **ktc_treeType_t tid_ParensDeclarator**
- const **ktc_treeType_t tid_ArrayDeclarator**
- const **ktc_treeType_t tid_FuncDeclarator**
- const **ktc_treeType_t tid_KRFuncDeclarator**
- const **ktc_treeType_t tid_Name**
- const **ktc_treeType_t tid_Dtor**
- const **ktc_treeType_t tid_SuffixFunc**
- const **ktc_treeType_t tid_OpFunc**
- const **ktc_treeType_t tid_ConvFunc**
- const **ktc_treeType_t tid_TemplateName**
- const **ktc_treeType_t tid_AnyPseudoDtor**
- const **ktc_treeType_t tid_UnparsedName**
- const **ktc_treeType_t tid_UsingDecl**
- const **ktc_treeType_t tid_UsingDirective**
- const **ktc_treeType_t tid_TypeAdjective**
- const **ktc_treeType_t tid_BuiltinType**
- const **ktc_treeType_t tid_TypeOfExpr**
- const **ktc_treeType_t tid_TypeOfType**
- const **ktc_treeType_t tid_PseudoDtor**
- const **ktc_treeType_t tid_QualifiedPseudoDtor**
- const **ktc_childId_t cid_Next**
- const **ktc_childId_t cid_Decls**
- const **ktc_childId_t cid_Expr**
- const **ktc_childId_t cid_Stmt**
- const **ktc_childId_t cid_LambdaCapture**
- const **ktc_childId_t cid_CtorInit**
- const **ktc_childId_t cid_Handlers**
- const **ktc_childId_t cid_Exception**
- const **ktc_childId_t cid_DeclSpecs**
- const **ktc_childId_t cid_Declarator**
- const **ktc_childId_t cid_Attributes**
- const **ktc_childId_t cid_Designators**
- const **ktc_childId_t cid_Lower**
- const **ktc_childId_t cid_Upper**
- const **ktc_childId_t cid_Qualifier**
- const **ktc_childId_t cid_Name**
- const **ktc_childId_t cid_Params**
- const **ktc_childId_t cid_Throw**
- const **ktc_childId_t cid_TrailingReturnType**
- const **ktc_childId_t cid_Args**
- const **ktc_childId_t cid_MemberInitializers**
- const **ktc_childId_t cid_TypeIds**
- const **ktc_childId_t cid_Decl**

- const **ktc_childId_t cid_Type**
- const **ktc_childId_t cid_AttributeSpec**
- const **ktc_childId_t cid_PropertyFuncs**
- const **ktc_childId_t cid_Declarators**
- const **ktc_childId_t cid_FuncBody**
- const **ktc_childId_t cid_TemplateParams**
- const **ktc_childId_t cid_MemberDecl**
- const **ktc_childId_t cid_NameSpec**
- const **ktc_childId_t cid_CVQualifiers**
- const **ktc_childId_t cid_Bits**
- const **ktc_childId_t cid_Init**
- const **ktc_childId_t cid_Inits**
- const **ktc_childId_t cid_Index**
- const **ktc_childId_t cid_Adjacent**
- const **ktc_childId_t cid_Literal**
- const **ktc_childId_t cid_Func**
- const **ktc_childId_t cid_Left**
- const **ktc_childId_t cid_Right**
- const **ktc_childId_t cid_Cond**
- const **ktc_childId_t cid_Then**
- const **ktc_childId_t cid_Else**
- const **ktc_childId_t cid_Placement**
- const **ktc_childId_t cid_Initializer**
- const **ktc_childId_t cid_Introducer**
- const **ktc_childId_t cid_TemplateName**
- const **ktc_childId_t cid_KRParams**
- const **ktc_childId_t cid_Label**
- const **ktc_childId_t cid_Stmts**
- const **ktc_childId_t cid_Handler**
- const **ktc_childId_t cid_Default**
- const **ktc_childId_t cid_Base**
- const **ktc_childId_t cid_Enumerators**
- const **ktc_childId_t cid_AttributeSpecs**
- const **ktc_childId_t cid_BaseSpecs**
- const **ktc_childId_t cid_MemberDecls**
- const **ktc_childId_t cid_Size**
- const **ktc_childId_t cid_ConversionType**
- const int **KTC_OPCODE_NONE**
- const int **KTC_OPCODE_ASSIGN**
- const int **KTC_OPCODE_MULASSIGN**
- const int **KTC_OPCODE_DIVASSIGN**
- const int **KTC_OPCODE_MODASSIGN**
- const int **KTC_OPCODE_ADDASSIGN**
- const int **KTC_OPCODE_SUBASSIGN**
- const int **KTC_OPCODE_ASLASSIGN**
- const int **KTC_OPCODE_ASRASSIGN**
- const int **KTC_OPCODE_ANDASSIGN**
- const int **KTC_OPCODE_XORASSIGN**
- const int **KTC_OPCODE_ORASSIGN**
- const int **KTC_OPCODE_COMMA**
- const int **KTC_OPCODE_COND**
- const int **KTC_OPCODE_LOGOR**
- const int **KTC_OPCODE_LOGAND**
- const int **KTC_OPCODE_BITOR**
- const int **KTC_OPCODE_BITXOR**

- const int **KTC_OPCODE_BITAND**
- const int **KTC_OPCODE_EQ**
- const int **KTC_OPCODE_NE**
- const int **KTC_OPCODE_LT**
- const int **KTC_OPCODE_GT**
- const int **KTC_OPCODE_LE**
- const int **KTC_OPCODE_GE**
- const int **KTC_OPCODE_ASR**
- const int **KTC_OPCODE_ASL**
- const int **KTC_OPCODE_ADD**
- const int **KTC_OPCODE_SUB**
- const int **KTC_OPCODE_MUL**
- const int **KTC_OPCODE_DIV**
- const int **KTC_OPCODE_MOD**
- const int **KTC_OPCODE_PREINC**
- const int **KTC_OPCODE_PREDEC**
- const int **KTC_OPCODE_SIZEOF**
- const int **KTC_OPCODE_DEREF**
- const int **KTC_OPCODE_ADDRESS**
- const int **KTC_OPCODE_PLUS**
- const int **KTC_OPCODE_MINUS**
- const int **KTC_OPCODE_BITNOT**
- const int **KTC_OPCODE_LOGNOT**
- const int **KTC_OPCODE_POSTINC**
- const int **KTC_OPCODE_POSTDEC**
- const int **KTC_OPCODE_FIELD**
- const int **KTC_OPCODE_FIELDREF**
- const int **KTC_OPCODE_DOTAST**
- const int **KTC_OPCODE_DEREFAST**
- const int **KTC_OPCODE_ROUND_BRACKETS**
- const int **KTC_OPCODE_SQUARE_BRACKETS**
- const int **KTC_OPCODE_THROW**
- const int **KTC_OPCODE_MIN**
- const int **KTC_OPCODE_MAX**
- const int **KTC_STORAGECLASS_NONE**
- const int **KTC_STORAGECLASS_TYPEDEF**
- const int **KTC_STORAGECLASS_EXTERN**
- const int **KTC_STORAGECLASS_STATIC**
- const int **KTC_STORAGECLASS_AUTO**
- const int **KTC_STORAGECLASS_REGISTER**
- const int **KTC_STORAGECLASS_MUTABLE**
- const int **KTC_STORAGECLASS_THREADLOCAL**
- const int **KTC_CVQUALIFIER_NONE**
- const int **KTC_CVQUALIFIER_CONST**
- const int **KTC_CVQUALIFIER_VOLATILE**
- const int **KTC_CVQUALIFIER_RESTRICT**
- const int **KTC_CLASSTAG_NONE**
- const int **KTC_CLASSTAG_STRUCT**
- const int **KTC_CLASSTAG_UNION**
- const int **KTC_CLASSTAG_CLASS**
- const int **KTC_BUILTINTYPE_NONE**
- const int **KTC_BUILTINTYPE_VOID**
- const int **KTC_BUILTINTYPE_BOOL**
- const int **KTC_BUILTINTYPE_WCHAR_T**
- const int **KTC_BUILTINTYPE_CHAR**

- const int **KTC_BUILTINTYPE_SIGNEDCHAR**
- const int **KTC_BUILTINTYPE_UNSIGNEDCHAR**
- const int **KTC_BUILTINTYPE_SHORTINT**
- const int **KTC_BUILTINTYPE_SIGNEDSHORTINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDSHORTINT**
- const int **KTC_BUILTINTYPE_INT**
- const int **KTC_BUILTINTYPE_SIGNEDINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDINT**
- const int **KTC_BUILTINTYPE_LONGINT**
- const int **KTC_BUILTINTYPE_SIGNEDLONGINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDLONGINT**
- const int **KTC_BUILTINTYPE_LONGLONGINT**
- const int **KTC_BUILTINTYPE_SIGNEDLONGLONGINT**
- const int **KTC_BUILTINTYPE_UNSIGNEDLONGLONGINT**
- const int **KTC_BUILTINTYPE_FLOAT**
- const int **KTC_BUILTINTYPE_DOUBLE**
- const int **KTC_BUILTINTYPE_LONGDOUBLE**
- const int **KTC_BUILTINTYPE_NULLPTR_T**
- const int **KTC_FUNCSPECIFIER_NONE**
- const int **KTC_FUNCSPECIFIER_INLINE**
- const int **KTC_FUNCSPECIFIER_VIRTUAL**
- const int **KTC_FUNCSPECIFIER_EXPLICIT**
- const int **KTC_FUNCSPECIFIER_FRIEND**
- const int **KTC_CASTSPECIFIER_DYNAMIC**
- const int **KTC_CASTSPECIFIER_STATIC**
- const int **KTC_CASTSPECIFIER_REINTERPRET**
- const int **KTC_CASTSPECIFIER_CONST**
- const int **KTC_POINTEROPERATOR_NONE**
- const int **KTC_POINTEROPERATOR_POINTER**
- const int **KTC_POINTEROPERATOR_REFERENCE**
- const int **KTC_POINTEROPERATOR_RVALUE**

5.1.1 Function Documentation

5.1.1.1 ktc_is_AccessSpecification()

```
int ktc_is_AccessSpecification (
            ktc_tree_t t )
```

Checks that root of subtree has 'AccessSpecification' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AccessSpecification', 0 otherwise

5.1.1.2 ktc_is_AliasDecl()

```
int ktc_is_AliasDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'AliasDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AliasDecl', 0 otherwise

5.1.1.3 ktc_is_AlignAsExpr()

```
int ktc_is_AlignAsExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'AlignAsExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AlignAsExpr', 0 otherwise

5.1.1.4 ktc_is_AlignAsType()

```
int ktc_is_AlignAsType (
            ktc_tree_t t )
```

Checks that root of subtree has 'AlignAsType' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AlignAsType', 0 otherwise

5.1.1.5 ktc_is_AlignOfExpr()

```
int ktc_is_AlignOfExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'AlignOfExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AlignOfExpr', 0 otherwise

5.1.1.6 ktc_is_AnyAttribute()

```
int ktc_is_AnyAttribute (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyAttribute' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyAttribute', 0 otherwise

5.1.1.7 ktc_is_AnyCapture()

```
int ktc_is_AnyCapture (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyCapture' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyCapture', 0 otherwise

5.1.1.8 ktc_is_AnyDecl()

```
int ktc_is_AnyDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyDecl', 0 otherwise

5.1.1.9 ktc_is_AnyDeclarator()

```
int ktc_is_AnyDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyDeclarator', 0 otherwise

5.1.1.10 ktc_is_AnyDesignator()

```
int ktc_is_AnyDesignator (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyDesignator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyDesignator', 0 otherwise

5.1.1.11 ktc_is_AnyEnumerator()

```
int ktc_is_AnyEnumerator (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyEnumerator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyEnumerator', 0 otherwise

5.1.1.12 ktc_is_AnyExpr()

```
int ktc_is_AnyExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyExpr', 0 otherwise

5.1.1.13 ktc_is_AnyFuncBody()

```
int ktc_is_AnyFuncBody (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyFuncBody' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyFuncBody', 0 otherwise

5.1.1.14 ktc_is_AnyInitializer()

```
int ktc_is_AnyInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyInitializer', 0 otherwise

5.1.1.15 ktc_is_AnyLabel()

```
int ktc_is_AnyLabel (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyLabel' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyLabel', 0 otherwise

5.1.1.16 ktc_is_AnyMemberDecl()

```
int ktc_is_AnyMemberDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyMemberDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyMemberDecl', 0 otherwise

5.1.1.17 ktc_is_AnyName()

```
int ktc_is_AnyName (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyName', 0 otherwise

5.1.1.18 ktc_is_AnyNameQualifier()

```
int ktc_is_AnyNameQualifier (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyNameQualifier' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyNameQualifier', 0 otherwise

5.1.1.19 ktc_is_AnyNames()

```
int ktc_is_AnyNames (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyNames' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyNames', 0 otherwise

5.1.1.20 ktc_is_AnyNameSpec()

```
int ktc_is_AnyNameSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyNameSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyNameSpec', 0 otherwise

5.1.1.21 ktc_is_AnyNonPtrDeclarator()

```
int ktc_is_AnyNonPtrDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyNonPtrDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyNonPtrDeclarator', 0 otherwise

5.1.1.22 ktc_is_AnyParamName()

```
int ktc_is_AnyParamName (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyParamName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyParamName', 0 otherwise

5.1.1.23 ktc_is_AnyPropertyFunc()

```
int ktc_is_AnyPropertyFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyPropertyFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyPropertyFunc', 0 otherwise

5.1.1.24 ktc_is_AnyPseudoDtor()

```
int ktc_is_AnyPseudoDtor (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyPseudoDtor' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyPseudoDtor', 0 otherwise

5.1.1.25 ktc_is_AnyStmt()

```
int ktc_is_AnyStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyStmt', 0 otherwise

5.1.1.26 ktc_is_AnyTemplateArg()

```
int ktc_is_AnyTemplateArg (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyTemplateArg' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyTemplateArg', 0 otherwise

5.1.1.27 ktc_is_AnyTypeName()

```
int ktc_is_AnyTypeName (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyTypeName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyTypeName', 0 otherwise

5.1.1.28 ktc_is_AnyTypeOf()

```
int ktc_is_AnyTypeOf (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyTypeOf' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyTypeOf', 0 otherwise

5.1.1.29 ktc_is_AnyTypeParam()

```
int ktc_is_AnyTypeParam (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyTypeParam' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyTypeParam', 0 otherwise

5.1.1.30 ktc_is_AnyUsing()

```
int ktc_is_AnyUsing (
            ktc_tree_t t )
```

Checks that root of subtree has 'AnyUsing' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AnyUsing', 0 otherwise

5.1.1.31 ktc_is_ArrayDeclarator()

```
int ktc_is_ArrayDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'ArrayDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ArrayDeclarator', 0 otherwise

5.1.1.32 ktc_is_AsmDef()

```
int ktc_is_AsmDef (
            ktc_tree_t t )
```

Checks that root of subtree has 'AsmDef' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AsmDef', 0 otherwise

5.1.1.33 ktc_is_AsmStmt()

```
int ktc_is_AsmStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'AsmStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AsmStmt', 0 otherwise

5.1.1.34 ktc_is_Attribute()

```
int ktc_is_Attribute (
            ktc_tree_t t )
```

Checks that root of subtree has 'Attribute' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Attribute', 0 otherwise

5.1.1.35 ktc_is_AttributedDeclarator()

```
int ktc_is_AttributedDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'AttributedDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AttributedDeclarator', 0 otherwise

5.1.1.36 ktc_is_AttributeDeclSpec()

```
int ktc_is_AttributeDeclSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'AttributeDeclSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AttributeDeclSpec', 0 otherwise

5.1.1.37 ktc_is_Attributes()

```
int ktc_is_Attributes (
            ktc_tree_t t )
```

Checks that root of subtree has 'Attributes' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Attributes', 0 otherwise

5.1.1.38 ktc_is_AttributeSpec()

```
int ktc_is_AttributeSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'AttributeSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AttributeSpec', 0 otherwise

5.1.1.39 ktc_is_AttributeSpecs()

```
int ktc_is_AttributeSpecs (
            ktc_tree_t t )
```

Checks that root of subtree has 'AttributeSpecs' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AttributeSpecs', 0 otherwise

5.1.1.40 ktc_is_AttributeWithArgs()

```
int ktc_is_AttributeWithArgs (
            ktc_tree_t t )
```

Checks that root of subtree has 'AttributeWithArgs' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AttributeWithArgs', 0 otherwise

5.1.1.41 ktc_is_AutoType()

```
int ktc_is_AutoType (
            ktc_tree_t t )
```

Checks that root of subtree has 'AutoType' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'AutoType', 0 otherwise

5.1.1.42 ktc_is_BaseSpec()

```
int ktc_is_BaseSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'BaseSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'BaseSpec', 0 otherwise

5.1.1.43 ktc_is_BaseSpecs()

```
int ktc_is_BaseSpecs (
            ktc_tree_t t )
```

Checks that root of subtree has 'BaseSpecs' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'BaseSpecs', 0 otherwise

5.1.1.44 ktc_is_BinaryExpr()

```
int ktc_is_BinaryExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'BinaryExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'BinaryExpr', 0 otherwise

5.1.1.45 ktc_is_BitFieldDeclarator()

```
int ktc_is_BitFieldDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'BitFieldDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'BitFieldDeclarator', 0 otherwise

5.1.1.46 ktc_is_BoolLiteralExpr()

```
int ktc_is_BoolLiteralExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'BoolLiteralExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'BoolLiteralExpr', 0 otherwise

5.1.1.47 ktc_is_BreakStmt()

```
int ktc_is_BreakStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'BreakStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'BreakStmt', 0 otherwise

5.1.1.48 ktc_is_BuiltinType()

```
int ktc_is_BuiltinType (
            ktc_tree_t t )
```

Checks that root of subtree has 'BuiltinType' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'BuiltinType', 0 otherwise

5.1.1.49 ktc_is_CallExpr()

```
int ktc_is_CallExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'CallExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CallExpr', 0 otherwise

5.1.1.50 ktc_is_Capture()

```
int ktc_is_Capture (
            ktc_tree_t t )
```

Checks that root of subtree has 'Capture' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Capture', 0 otherwise

5.1.1.51 ktc_is_CaptureDefault()

```
int ktc_is_CaptureDefault (
            ktc_tree_t t )
```

Checks that root of subtree has 'CaptureDefault' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CaptureDefault', 0 otherwise

5.1.1.52 ktc_is_CaseLabel()

```
int ktc_is_CaseLabel (
            ktc_tree_t t )
```

Checks that root of subtree has 'CaseLabel' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CaseLabel', 0 otherwise

5.1.1.53 ktc_is_CaseRangeLabel()

```
int ktc_is_CaseRangeLabel (
            ktc_tree_t t )
```

Checks that root of subtree has 'CaseRangeLabel' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CaseRangeLabel', 0 otherwise

5.1.1.54 ktc_is_CastExpr()

```
int ktc_is_CastExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'CastExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CastExpr', 0 otherwise

5.1.1.55 ktc_is_ClassType()

```
int ktc_is_ClassType (
            ktc_tree_t t )
```

Checks that root of subtree has 'ClassType' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ClassType', 0 otherwise

5.1.1.56 ktc_is_CompoundStmt()

```
int ktc_is_CompoundStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'CompoundStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CompoundStmt', 0 otherwise

5.1.1.57 ktc_is_ConditionalExpr()

```
int ktc_is_ConditionalExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'ConditionalExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ConditionalExpr', 0 otherwise

5.1.1.58 ktc_is_ConstExpr()

```
int ktc_is_ConstExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'ConstExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ConstExpr', 0 otherwise

5.1.1.59 ktc_is_ContinueStmt()

```
int ktc_is_ContinueStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'ContinueStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ContinueStmt', 0 otherwise

5.1.1.60 ktc_is_ConvFunc()

```
int ktc_is_ConvFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'ConvFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ConvFunc', 0 otherwise

5.1.1.61 ktc_is_CopyInitializer()

```
int ktc_is_CopyInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'CopyInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CopyInitializer', 0 otherwise

5.1.1.62 ktc_is_CtorInitializer()

```
int ktc_is_CtorInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'CtorInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CtorInitializer', 0 otherwise

5.1.1.63 ktc_is_CVQualifier()

```
int ktc_is_CVQualifier (
            ktc_tree_t t )
```

Checks that root of subtree has 'CVQualifier' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'CVQualifier', 0 otherwise

5.1.1.64 ktc_is_Decl()

```
int ktc_is_Decl (
            ktc_tree_t t )
```

Checks that root of subtree has 'Decl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Decl', 0 otherwise

5.1.1.65 ktc_is_DeclEllipsis()

```
int ktc_is_DeclEllipsis (
            ktc_tree_t t )
```

Checks that root of subtree has 'DeclEllipsis' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DeclEllipsis', 0 otherwise

5.1.1.66 ktc_is_DeclOrStmt()

```
int ktc_is_DeclOrStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'DeclOrStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DeclOrStmt', 0 otherwise

5.1.1.67 ktc_is_DeclOrStmts()

```
int ktc_is_DeclOrStmts (
            ktc_tree_t t )
```

Checks that root of subtree has 'DeclOrStmts' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DeclOrStmts', 0 otherwise

5.1.1.68 ktc_is_DeclSpec()

```
int ktc_is_DeclSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'DeclSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DeclSpec', 0 otherwise

5.1.1.69 ktc_is_DeclSpecs()

```
int ktc_is_DeclSpecs (
            ktc_tree_t t )
```

Checks that root of subtree has 'DeclSpecs' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DeclSpecs', 0 otherwise

5.1.1.70 ktc_is_DefaultException()

```
int ktc_is_DefaultException (
            ktc_tree_t t )
```

Checks that root of subtree has 'DefaultException' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DefaultException', 0 otherwise

5.1.1.71 ktc_is_DefaultLabel()

```
int ktc_is_DefaultLabel (
            ktc_tree_t t )
```

Checks that root of subtree has 'DefaultLabel' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DefaultLabel', 0 otherwise

5.1.1.72 ktc_is_DeleteExpr()

```
int ktc_is_DeleteExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'DeleteExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DeleteExpr', 0 otherwise

5.1.1.73 ktc_is_DenyThrowSpec()

```
int ktc_is_DenyThrowSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'DenyThrowSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DenyThrowSpec', 0 otherwise

5.1.1.74 ktc_is_Designators()

```
int ktc_is_Designators (
            ktc_tree_t t )
```

Checks that root of subtree has 'Designators' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Designators', 0 otherwise

5.1.1.75 ktc_is_DirectInitializer()

```
int ktc_is_DirectInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'DirectInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DirectInitializer', 0 otherwise

5.1.1.76 ktc_is_DoDeclStmt()

```
int ktc_is_DoDeclStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'DoDeclStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DoDeclStmt', 0 otherwise

5.1.1.77 ktc_is_DoStmt()

```
int ktc_is_DoStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'DoStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'DoStmt', 0 otherwise

5.1.1.78 ktc_is_Dtor()

```
int ktc_is_Dtor (
            ktc_tree_t t )
```

Checks that root of subtree has 'Dtor' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Dtor', 0 otherwise

5.1.1.79 ktc_is_Enumerator()

```
int ktc_is_Enumerator (
            ktc_tree_t t )
```

Checks that root of subtree has 'Enumerator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Enumerator', 0 otherwise

5.1.1.80 ktc_is_Enumerators()

```
int ktc_is_Enumerators (
            ktc_tree_t t )
```

Checks that root of subtree has 'Enumerators' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Enumerators', 0 otherwise

5.1.1.81 ktc_is_EnumType()

```
int ktc_is_EnumType (
            ktc_tree_t t )
```

Checks that root of subtree has 'EnumType' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'EnumType', 0 otherwise

5.1.1.82 ktc_is_ExceptHandler()

```
int ktc_is_ExceptHandler (
            ktc_tree_t t )
```

Checks that root of subtree has 'ExceptHandler' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ExceptHandler', 0 otherwise

5.1.1.83 ktc_is_Exception()

```
int ktc_is_Exception (
            ktc_tree_t t )
```

Checks that root of subtree has 'Exception' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Exception', 0 otherwise

5.1.1.84 ktc_is_ExceptionSpec()

```
int ktc_is_ExceptionSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'ExceptionSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ExceptionSpec', 0 otherwise

5.1.1.85 ktc_is_ExplicitInstantiation()

```
int ktc_is_ExplicitInstantiation (
            ktc_tree_t t )
```

Checks that root of subtree has 'ExplicitInstantiation' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ExplicitInstantiation', 0 otherwise

5.1.1.86 ktc_is_ExprArg()

```
int ktc_is_ExprArg (
            ktc_tree_t t )
```

Checks that root of subtree has 'ExprArg' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ExprArg', 0 otherwise

5.1.1.87 ktc_is_Exprs()

```
int ktc_is_Exprs (
            ktc_tree_t t )
```

Checks that root of subtree has 'Exprs' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Exprs', 0 otherwise

5.1.1.88 ktc_is_ExprStmt()

```
int ktc_is_ExprStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'ExprStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ExprStmt', 0 otherwise

5.1.1.89 ktc_is_ExprTypeIdExpr()

```
int ktc_is_ExprTypeIdExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'ExprTypeIdExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ExprTypeIdExpr', 0 otherwise

5.1.1.90 ktc_is_FieldDesignator()

```
int ktc_is_FieldDesignator (
            ktc_tree_t t )
```

Checks that root of subtree has 'FieldDesignator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'FieldDesignator', 0 otherwise

5.1.1.91 ktc_is_FinallyHandler()

```
int ktc_is_FinallyHandler (
            ktc_tree_t t )
```

Checks that root of subtree has 'FinallyHandler' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'FinallyHandler', 0 otherwise

5.1.1.92 ktc_is_ForEachStmt()

```
int ktc_is_ForEachStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'ForEachStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ForEachStmt', 0 otherwise

5.1.1.93 ktc_is_ForRangeStmt()

```
int ktc_is_ForRangeStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'ForRangeStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ForRangeStmt', 0 otherwise

5.1.1.94 ktc_is_ForStmt()

```
int ktc_is_ForStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'ForStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ForStmt', 0 otherwise

5.1.1.95 ktc_is_FuncBody()

```
int ktc_is_FuncBody (
            ktc_tree_t t )
```

Checks that root of subtree has 'FuncBody' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'FuncBody', 0 otherwise

5.1.1.96 ktc_is_FuncDeclarator()

```
int ktc_is_FuncDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'FuncDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'FuncDeclarator', 0 otherwise

5.1.1.97 ktc_is_FuncDef()

```
int ktc_is_FuncDef (
            ktc_tree_t t )
```

Checks that root of subtree has 'FuncDef' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'FuncDef', 0 otherwise

5.1.1.98 ktc_is_FuncSpec()

```
int ktc_is_FuncSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'FuncSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'FuncSpec', 0 otherwise

5.1.1.99 ktc_is_FuncTryBlock()

```
int ktc_is_FuncTryBlock (
            ktc_tree_t t )
```

Checks that root of subtree has 'FuncTryBlock' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'FuncTryBlock', 0 otherwise

5.1.1.100 ktc_is_GenericAttribute()

```
int ktc_is_GenericAttribute (
            ktc_tree_t t )
```

Checks that root of subtree has 'GenericAttribute' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'GenericAttribute', 0 otherwise

5.1.1.101 ktc_is_GlobalScope()

```
int ktc_is_GlobalScope (
            ktc_tree_t t )
```

Checks that root of subtree has 'GlobalScope' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'GlobalScope', 0 otherwise

5.1.1.102 ktc_is_GotoStmt()

```
int ktc_is_GotoStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'GotoStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'GotoStmt', 0 otherwise

5.1.1.103 ktc_is_Handler()

```
int ktc_is_Handler (
            ktc_tree_t t )
```

Checks that root of subtree has 'Handler' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Handler', 0 otherwise

5.1.1.104 ktc_is_Handlers()

```
int ktc_is_Handlers (
            ktc_tree_t t )
```

Checks that root of subtree has 'Handlers' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Handlers', 0 otherwise

5.1.1.105 ktc_is_IdExpr()

```
int ktc_is_IdExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'IdExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'IdExpr', 0 otherwise

5.1.1.106 ktc_is_IfDeclStmt()

```
int ktc_is_IfDeclStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'IfDeclStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'IfDeclStmt', 0 otherwise

5.1.1.107 ktc_is_IfStmt()

```
int ktc_is_IfStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'IfStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'IfStmt', 0 otherwise

5.1.1.108 ktc_is_IndexDesignator()

```
int ktc_is_IndexDesignator (
            ktc_tree_t t )
```

Checks that root of subtree has 'IndexDesignator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'IndexDesignator', 0 otherwise

5.1.1.109 ktc_is_IndexExpr()

```
int ktc_is_IndexExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'IndexExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'IndexExpr', 0 otherwise

5.1.1.110 ktc_is_InitClause()

```
int ktc_is_InitClause (
            ktc_tree_t t )
```

Checks that root of subtree has 'InitClause' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'InitClause', 0 otherwise

5.1.1.111 ktc_is_InitializedDeclarator()

```
int ktc_is_InitializedDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'InitializedDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'InitializedDeclarator', 0 otherwise

5.1.1.112 ktc_is_InitializerExpr()

```
int ktc_is_InitializerExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'InitializerExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'InitializerExpr', 0 otherwise

5.1.1.113 ktc_is_Initializers()

```
int ktc_is_Initializers (
            ktc_tree_t t )
```

Checks that root of subtree has 'Initializers' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Initializers', 0 otherwise

5.1.1.114 ktc_is_KRFuncDeclarator()

```
int ktc_is_KRFuncDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'KRFuncDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'KRFuncDeclarator', 0 otherwise

5.1.1.115 ktc_is_Label()

```
int ktc_is_Label (
            ktc_tree_t t )
```

Checks that root of subtree has 'Label' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Label', 0 otherwise

5.1.1.116 ktc_is_LabeledStmt()

```
int ktc_is_LabeledStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'LabeledStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'LabeledStmt', 0 otherwise

5.1.1.117 ktc_is_LambdaDeclarator()

```
int ktc_is_LambdaDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'LambdaDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'LambdaDeclarator', 0 otherwise

5.1.1.118 ktc_is_LambdaExpr()

```
int ktc_is_LambdaExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'LambdaExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'LambdaExpr', 0 otherwise

5.1.1.119 ktc_is_LambdaIntroducer()

```
int ktc_is_LambdaIntroducer (
            ktc_tree_t t )
```

Checks that root of subtree has 'LambdaIntroducer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'LambdaIntroducer', 0 otherwise

5.1.1.120 ktc_is_LeaveStmt()

```
int ktc_is_LeaveStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'LeaveStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'LeaveStmt', 0 otherwise

5.1.1.121 ktc_is_LinkageSpec()

```
int ktc_is_LinkageSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'LinkageSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'LinkageSpec', 0 otherwise

5.1.1.122 ktc_is_LiteralExpr()

```
int ktc_is_LiteralExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'LiteralExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'LiteralExpr', 0 otherwise

5.1.1.123 ktc_is_MaybeCtorInitializer()

```
int ktc_is_MaybeCtorInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'MaybeCtorInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MaybeCtorInitializer', 0 otherwise

5.1.1.124 ktc_is_MaybeDeclarator()

```
int ktc_is_MaybeDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'MaybeDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MaybeDeclarator', 0 otherwise

5.1.1.125 ktc_is_MaybeException()

```
int ktc_is_MaybeException (
            ktc_tree_t t )
```

Checks that root of subtree has 'MaybeException' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MaybeException', 0 otherwise

5.1.1.126 ktc_is_MaybeExceptionSpec()

```
int ktc_is_MaybeExceptionSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'MaybeExceptionSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MaybeExceptionSpec', 0 otherwise

5.1.1.127 ktc_is_MaybeLambdaDeclarator()

```
int ktc_is_MaybeLambdaDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'MaybeLambdaDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MaybeLambdaDeclarator', 0 otherwise

5.1.1.128 ktc_is_MaybeNewInitializer()

```
int ktc_is_MaybeNewInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'MaybeNewInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MaybeNewInitializer', 0 otherwise

5.1.1.129 ktc_is_MaybeTypeId()

```
int ktc_is_MaybeTypeId (
            ktc_tree_t t )
```

Checks that root of subtree has 'MaybeTypeId' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MaybeTypeId', 0 otherwise

5.1.1.130 ktc_is_MemberDecl()

```
int ktc_is_MemberDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberDecl', 0 otherwise

5.1.1.131 ktc_is_MemberDecls()

```
int ktc_is_MemberDecls (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberDecls' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberDecls', 0 otherwise

5.1.1.132 ktc_is_MemberDesignator()

```
int ktc_is_MemberDesignator (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberDesignator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberDesignator', 0 otherwise

5.1.1.133 ktc_is_MemberExpr()

```
int ktc_is_MemberExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberExpr', 0 otherwise

5.1.1.134 ktc_is_MemberFunc()

```
int ktc_is_MemberFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberFunc', 0 otherwise

5.1.1.135 ktc_is_MemberInitializer()

```
int ktc_is_MemberInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberInitializer', 0 otherwise

5.1.1.136 ktc_is_MemberInitializers()

```
int ktc_is_MemberInitializers (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberInitializers' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberInitializers', 0 otherwise

5.1.1.137 ktc_is_MemberTemplate()

```
int ktc_is_MemberTemplate (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberTemplate' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberTemplate', 0 otherwise

5.1.1.138 ktc_is_MemberUsingDecl()

```
int ktc_is_MemberUsingDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'MemberUsingDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'MemberUsingDecl', 0 otherwise

5.1.1.139 ktc_is_Name()

```
int ktc_is_Name (
            ktc_tree_t t )
```

Checks that root of subtree has 'Name' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Name', 0 otherwise

5.1.1.140 ktc_is_NameDeclarator()

```
int ktc_is_NameDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'NameDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NameDeclarator', 0 otherwise

5.1.1.141 ktc_is_NamespaceAlias()

```
int ktc_is_NamespaceAlias (
            ktc_tree_t t )
```

Checks that root of subtree has 'NamespaceAlias' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NamespaceAlias', 0 otherwise

5.1.1.142 ktc_is_NamespaceDecl()

```
int ktc_is_NamespaceDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'NamespaceDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NamespaceDecl', 0 otherwise

5.1.1.143 ktc_is_NameSpec()

```
int ktc_is_NameSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'NameSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NameSpec', 0 otherwise

5.1.1.144 ktc_is_NewExpr()

```
int ktc_is_NewExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'NewExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NewExpr', 0 otherwise

5.1.1.145 ktc_is_NewInitializer()

```
int ktc_is_NewInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'NewInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NewInitializer', 0 otherwise

5.1.1.146 ktc_is_NoAttribute()

```
int ktc_is_NoAttribute (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoAttribute' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoAttribute', 0 otherwise

5.1.1.147 ktc_is_NoAttributeSpec()

```
int ktc_is_NoAttributeSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoAttributeSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoAttributeSpec', 0 otherwise

5.1.1.148 ktc_is_NoBaseSpec()

```
int ktc_is_NoBaseSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoBaseSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoBaseSpec', 0 otherwise

5.1.1.149 ktc_is_NoCapture()

```
int ktc_is_NoCapture (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoCapture' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoCapture', 0 otherwise

5.1.1.150 ktc_is_NoCtorInitializer()

```
int ktc_is_NoCtorInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoCtorInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoCtorInitializer', 0 otherwise

5.1.1.151 ktc_is_Node()

```
int ktc_is_Node (
            ktc_tree_t t )
```

Checks that root of subtree has 'Node' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Node', 0 otherwise

5.1.1.152 ktc_is_NoDeclarator()

```
int ktc_is_NoDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoDeclarator', 0 otherwise

5.1.1.153 ktc_is_NoDeclOrStmt()

```
int ktc_is_NoDeclOrStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoDeclOrStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoDeclOrStmt', 0 otherwise

5.1.1.154 ktc_is_NoDeclSpec()

```
int ktc_is_NoDeclSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoDeclSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoDeclSpec', 0 otherwise

5.1.1.155 ktc_is_NoDesignator()

```
int ktc_is_NoDesignator (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoDesignator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoDesignator', 0 otherwise

5.1.1.156 ktc_is_NoEnumerator()

```
int ktc_is_NoEnumerator (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoEnumerator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoEnumerator', 0 otherwise

5.1.1.157 ktc_is_NoException()

```
int ktc_is_NoException (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoException' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoException', 0 otherwise

5.1.1.158 ktc_is_NoExceptionSpec()

```
int ktc_is_NoExceptionSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoExceptionSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoExceptionSpec', 0 otherwise

5.1.1.159 ktc_is_NoExpr()

```
int ktc_is_NoExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoExpr', 0 otherwise

5.1.1.160 ktc_is_NoHandler()

```
int ktc_is_NoHandler (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoHandler' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoHandler', 0 otherwise

5.1.1.161 ktc_is_NoInitializer()

```
int ktc_is_NoInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoInitializer', 0 otherwise

5.1.1.162 ktc_is_NoLambdaDeclarator()

```
int ktc_is_NoLambdaDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoLambdaDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoLambdaDeclarator', 0 otherwise

5.1.1.163 ktc_is_NoMemberDecl()

```
int ktc_is_NoMemberDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoMemberDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoMemberDecl', 0 otherwise

5.1.1.164 ktc_is_NoMemberInitializer()

```
int ktc_is_NoMemberInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoMemberInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoMemberInitializer', 0 otherwise

5.1.1.165 ktc_is_NoName()

```
int ktc_is_NoName (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoName', 0 otherwise

5.1.1.166 ktc_is_NoNameQualifier()

```
int ktc_is_NoNameQualifier (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoNameQualifier' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoNameQualifier', 0 otherwise

5.1.1.167 ktc_is_NoNewInitializer()

```
int ktc_is_NoNewInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoNewInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoNewInitializer', 0 otherwise

5.1.1.168 ktc_is_NoParamName()

```
int ktc_is_NoParamName (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoParamName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoParamName', 0 otherwise

5.1.1.169 ktc_is_NoPropertyFunc()

```
int ktc_is_NoPropertyFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoPropertyFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoPropertyFunc', 0 otherwise

5.1.1.170 ktc_is_NoTemplateArg()

```
int ktc_is_NoTemplateArg (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoTemplateArg' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoTemplateArg', 0 otherwise

5.1.1.171 ktc_is_NoTemplateParam()

```
int ktc_is_NoTemplateParam (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoTemplateParam' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoTemplateParam', 0 otherwise

5.1.1.172 ktc_is_NoTypeId()

```
int ktc_is_NoTypeId (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoTypeId' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoTypeId', 0 otherwise

5.1.1.173 ktc_is_NullptrLiteralExpr()

```
int ktc_is_NullptrLiteralExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'NullptrLiteralExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NullptrLiteralExpr', 0 otherwise

5.1.1.174 ktc_is_OpFunc()

```
int ktc_is_OpFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'OpFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'OpFunc', 0 otherwise

5.1.1.175 ktc_is_Param()

```
int ktc_is_Param (
            ktc_tree_t t )
```

Checks that root of subtree has 'Param' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Param', 0 otherwise

5.1.1.176 ktc_is_ParamName()

```
int ktc_is_ParamName (
            ktc_tree_t t )
```

Checks that root of subtree has 'ParamName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ParamName', 0 otherwise

5.1.1.177 ktc_is_ParamNames()

```
int ktc_is_ParamNames (
            ktc_tree_t t )
```

Checks that root of subtree has 'ParamNames' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ParamNames', 0 otherwise

5.1.1.178 ktc_is_ParensDeclarator()

```
int ktc_is_ParensDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'ParensDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ParensDeclarator', 0 otherwise

5.1.1.179 ktc_is_ParensExpr()

```
int ktc_is_ParensExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'ParensExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ParensExpr', 0 otherwise

5.1.1.180 ktc_is_PromisedFuncBody()

```
int ktc_is_PromisedFuncBody (
            ktc_tree_t t )
```

Checks that root of subtree has 'PromisedFuncBody' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PromisedFuncBody', 0 otherwise

5.1.1.181 ktc_is_PromisedMemberDecl()

```
int ktc_is_PromisedMemberDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'PromisedMemberDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PromisedMemberDecl', 0 otherwise

5.1.1.182 ktc_is_PropertyAttribute()

```
int ktc_is_PropertyAttribute (
            ktc_tree_t t )
```

Checks that root of subtree has 'PropertyAttribute' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PropertyAttribute', 0 otherwise

5.1.1.183 ktc_is_PropertyFuncs()

```
int ktc_is_PropertyFuncs (
            ktc_tree_t t )
```

Checks that root of subtree has 'PropertyFuncs' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PropertyFuncs', 0 otherwise

5.1.1.184 ktc_is_PropertyGetFunc()

```
int ktc_is_PropertyGetFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'PropertyGetFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PropertyGetFunc', 0 otherwise

5.1.1.185 ktc_is_PropertyPutFunc()

```
int ktc_is_PropertyPutFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'PropertyPutFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PropertyPutFunc', 0 otherwise

5.1.1.186 ktc_is_PseudoDtor()

```
int ktc_is_PseudoDtor (
            ktc_tree_t t )
```

Checks that root of subtree has 'PseudoDtor' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PseudoDtor', 0 otherwise

5.1.1.187 ktc_is_PtrDeclarator()

```
int ktc_is_PtrDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'PtrDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'PtrDeclarator', 0 otherwise

5.1.1.188 ktc_is_QualifiedName()

```
int ktc_is_QualifiedName (
            ktc_tree_t t )
```

Checks that root of subtree has 'QualifiedName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'QualifiedName', 0 otherwise

5.1.1.189 ktc_is_QualifiedPseudoDtor()

```
int ktc_is_QualifiedPseudoDtor (
            ktc_tree_t t )
```

Checks that root of subtree has 'QualifiedPseudoDtor' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'QualifiedPseudoDtor', 0 otherwise

5.1.1.190 ktc_is_RangeDesignator()

```
int ktc_is_RangeDesignator (
            ktc_tree_t t )
```

Checks that root of subtree has 'RangeDesignator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'RangeDesignator', 0 otherwise

5.1.1.191 ktc_is_ReservedTypeSpec()

```
int ktc_is_ReservedTypeSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'ReservedTypeSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ReservedTypeSpec', 0 otherwise

5.1.1.192 ktc_is_ReturnStmt()

```
int ktc_is_ReturnStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'ReturnStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ReturnStmt', 0 otherwise

5.1.1.193 ktc_is_SizeOfExpr()

```
int ktc_is_SizeOfExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'SizeOfExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'SizeOfExpr', 0 otherwise

5.1.1.194 ktc_is_SpecialCastExpr()

```
int ktc_is_SpecialCastExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'SpecialCastExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'SpecialCastExpr', 0 otherwise

5.1.1.195 ktc_is_StaticAssertDecl()

```
int ktc_is_StaticAssertDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'StaticAssertDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'StaticAssertDecl', 0 otherwise

5.1.1.196 ktc_is_StmtExpr()

```
int ktc_is_StmtExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'StmtExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'StmtExpr', 0 otherwise

5.1.1.197 ktc_is_StorageClass()

```
int ktc_is_StorageClass (
            ktc_tree_t t )
```

Checks that root of subtree has 'StorageClass' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'StorageClass', 0 otherwise

5.1.1.198 ktc_is_StringLiteralExpr()

```
int ktc_is_StringLiteralExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'StringLiteralExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'StringLiteralExpr', 0 otherwise

5.1.1.199 ktc_is_SuffixFunc()

```
int ktc_is_SuffixFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'SuffixFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'SuffixFunc', 0 otherwise

5.1.1.200 ktc_is_SuperScope()

```
int ktc_is_SuperScope (
            ktc_tree_t t )
```

Checks that root of subtree has 'SuperScope' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'SuperScope', 0 otherwise

5.1.1.201 ktc_is_SwitchDeclStmt()

```
int ktc_is_SwitchDeclStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'SwitchDeclStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'SwitchDeclStmt', 0 otherwise

5.1.1.202 ktc_is_SwitchStmt()

```
int ktc_is_SwitchStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'SwitchStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'SwitchStmt', 0 otherwise

5.1.1.203 ktc_is_TemplateArgs()

```
int ktc_is_TemplateArgs (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateArgs' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateArgs', 0 otherwise

5.1.1.204 ktc_is_TemplateDecl()

```
int ktc_is_TemplateDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateDecl', 0 otherwise

5.1.1.205 ktc_is_TemplateName()

```
int ktc_is_TemplateName (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateName', 0 otherwise

5.1.1.206 ktc_is_TemplateParam()

```
int ktc_is_TemplateParam (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateParam' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateParam', 0 otherwise

5.1.1.207 ktc_is_TemplateParams()

```
int ktc_is_TemplateParams (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateParams' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateParams', 0 otherwise

5.1.1.208 ktc_is_TemplateSpec()

```
int ktc_is_TemplateSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateSpec', 0 otherwise

5.1.1.209 ktc_is_TemplateTypeArg()

```
int ktc_is_TemplateTypeArg (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateTypeArg' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateTypeArg', 0 otherwise

5.1.1.210 ktc_is_TemplateTypeParam()

```
int ktc_is_TemplateTypeParam (
            ktc_tree_t t )
```

Checks that root of subtree has 'TemplateTypeParam' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TemplateTypeParam', 0 otherwise

5.1.1.211 ktc_is_ThisExpr()

```
int ktc_is_ThisExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'ThisExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ThisExpr', 0 otherwise

5.1.1.212 ktc_is_ThrowExpr()

```
int ktc_is_ThrowExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'ThrowExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'ThrowExpr', 0 otherwise

5.1.1.213 ktc_is_TranslationUnit()

```
int ktc_is_TranslationUnit (
            ktc_tree_t t )
```

Checks that root of subtree has 'TranslationUnit' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TranslationUnit', 0 otherwise

5.1.1.214 ktc_is_TruncatedInitClause()

```
int ktc_is_TruncatedInitClause (
            ktc_tree_t t )
```

Checks that root of subtree has 'TruncatedInitClause' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TruncatedInitClause', 0 otherwise

5.1.1.215 ktc_is_TryExceptStmt()

```
int ktc_is_TryExceptStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'TryExceptStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TryExceptStmt', 0 otherwise

5.1.1.216 ktc_is_TryFinallyStmt()

```
int ktc_is_TryFinallyStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'TryFinallyStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TryFinallyStmt', 0 otherwise

5.1.1.217 ktc_is_TryStmt()

```
int ktc_is_TryStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'TryStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TryStmt', 0 otherwise

5.1.1.218 ktc_is_TypeAdjective()

```
int ktc_is_TypeAdjective (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeAdjective' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeAdjective', 0 otherwise

5.1.1.219 ktc_is_TypeArg()

```
int ktc_is_TypeArg (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeArg' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeArg', 0 otherwise

5.1.1.220 ktc_is_TypeConvExpr()

```
int ktc_is_TypeConvExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeConvExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeConvExpr', 0 otherwise

5.1.1.221 ktc_is_TypeId()

```
int ktc_is_TypeId (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeId' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeId', 0 otherwise

5.1.1.222 ktc_is_TypeName()

```
int ktc_is_TypeName (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeName', 0 otherwise

5.1.1.223 ktc_is_TypeOfExpr()

```
int ktc_is_TypeOfExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeOfExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeOfExpr', 0 otherwise

5.1.1.224 ktc_is_TypeOfSpec()

```
int ktc_is_TypeOfSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeOfSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeOfSpec', 0 otherwise

5.1.1.225 ktc_is_TypeOfType()

```
int ktc_is_TypeOfType (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeOfType' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeOfType', 0 otherwise

5.1.1.226 ktc_is_TypeParam()

```
int ktc_is_TypeParam (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeParam' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeParam', 0 otherwise

5.1.1.227 ktc_is_TypeTypeIdExpr()

```
int ktc_is_TypeTypeIdExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'TypeTypeIdExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'TypeTypeIdExpr', 0 otherwise

5.1.1.228 ktc_is_UnaryExpr()

```
int ktc_is_UnaryExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnaryExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnaryExpr', 0 otherwise

5.1.1.229 ktc_is_UnparsedDecl()

```
int ktc_is_UnparsedDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedDecl', 0 otherwise

5.1.1.230 ktc_is_UnparsedDeclarator()

```
int ktc_is_UnparsedDeclarator (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedDeclarator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedDeclarator', 0 otherwise

5.1.1.231 ktc_is_UnparsedDeclSpec()

```
int ktc_is_UnparsedDeclSpec (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedDeclSpec' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedDeclSpec', 0 otherwise

5.1.1.232 ktc_is_UnparsedEnumerator()

```
int ktc_is_UnparsedEnumerator (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedEnumerator' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedEnumerator', 0 otherwise

5.1.1.233 ktc_is_UnparsedException()

```
int ktc_is_UnparsedException (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedException' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedException', 0 otherwise

5.1.1.234 ktc_is_UnparsedExpr()

```
int ktc_is_UnparsedExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedExpr', 0 otherwise

5.1.1.235 ktc_is_UnparsedInitializer()

```
int ktc_is_UnparsedInitializer (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedInitializer' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedInitializer', 0 otherwise

5.1.1.236 ktc_is_UnparsedLabel()

```
int ktc_is_UnparsedLabel (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedLabel' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedLabel', 0 otherwise

5.1.1.237 ktc_is_UnparsedMemberDecl()

```
int ktc_is_UnparsedMemberDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedMemberDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedMemberDecl', 0 otherwise

5.1.1.238 ktc_is_UnparsedName()

```
int ktc_is_UnparsedName (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedName', 0 otherwise

5.1.1.239 ktc_is_UnparsedNameQualifier()

```
int ktc_is_UnparsedNameQualifier (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedNameQualifier' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedNameQualifier', 0 otherwise

5.1.1.240 ktc_is_UnparsedParamName()

```
int ktc_is_UnparsedParamName (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedParamName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedParamName', 0 otherwise

5.1.1.241 ktc_is_UnparsedPropertyFunc()

```
int ktc_is_UnparsedPropertyFunc (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedPropertyFunc' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedPropertyFunc', 0 otherwise

5.1.1.242 ktc_is_UnparsedStmt()

```
int ktc_is_UnparsedStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnparsedStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnparsedStmt', 0 otherwise

5.1.1.243 ktc_is_UnqualifiedName()

```
int ktc_is_UnqualifiedName (
            ktc_tree_t t )
```

Checks that root of subtree has 'UnqualifiedName' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UnqualifiedName', 0 otherwise

5.1.1.244 ktc_is_UserLiteralExpr()

```
int ktc_is_UserLiteralExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'UserLiteralExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UserLiteralExpr', 0 otherwise

5.1.1.245 ktc_is_UserStringLiteralExpr()

```
int ktc_is_UserStringLiteralExpr (
            ktc_tree_t t )
```

Checks that root of subtree has 'UserStringLiteralExpr' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UserStringLiteralExpr', 0 otherwise

5.1.1.246 ktc_is_UsingDecl()

```
int ktc_is_UsingDecl (
            ktc_tree_t t )
```

Checks that root of subtree has 'UsingDecl' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UsingDecl', 0 otherwise

5.1.1.247 ktc_is_UsingDirective()

```
int ktc_is_UsingDirective (
            ktc_tree_t t )
```

Checks that root of subtree has 'UsingDirective' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'UsingDirective', 0 otherwise

5.1.1.248 ktc_is_WhileDeclStmt()

```
int ktc_is_WhileDeclStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'WhileDeclStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'WhileDeclStmt', 0 otherwise

5.1.1.249 ktc_is_WhileStmt()

```
int ktc_is_WhileStmt (
            ktc_tree_t t )
```

Checks that root of subtree has 'WhileStmt' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'WhileStmt', 0 otherwise

5.1.2 Variable Documentation

5.1.2.1 cid_Adjacent

```
const ktc_childId_t cid_Adjacent
```

Constant to identify 'Adjacent' child of AST nodes

5.1.2.2 cid_Args

```
const ktc_childId_t cid_Args
```

Constant to identify 'Args' child of AST nodes

5.1.2.3 cid_Attributes

```
const ktc_childId_t cid_Attributes
```

Constant to identify 'Attributes' child of AST nodes

5.1.2.4 cid_AttributeSpec

```
const ktc_childId_t cid_AttributeSpec
```

Constant to identify 'AttributeSpec' child of AST nodes

5.1.2.5 cid_AttributeSpecs

```
const ktc_childId_t cid_AttributeSpecs
```

Constant to identify 'AttributeSpecs' child of AST nodes

5.1.2.6 cid_Base

```
const ktc_childId_t cid_Base
```

Constant to identify 'Base' child of AST nodes

5.1.2.7 cid_BaseSpecs

```
const ktc_childId_t cid_BaseSpecs
```

Constant to identify 'BaseSpecs' child of AST nodes

5.1.2.8 cid_Bits

```
const ktc_childId_t cid_Bits
```

Constant to identify 'Bits' child of AST nodes

5.1.2.9 cid_Cond

```
const ktc_childId_t cid_Cond
```

Constant to identify 'Cond' child of AST nodes

5.1.2.10 cid_ConversionType

```
const ktc_childId_t cid_ConversionType
```

Constant to identify 'ConversionType' child of AST nodes

5.1.2.11 cid_CtorInit

```
const ktc_childId_t cid_CtorInit
```

Constant to identify 'CtorInit' child of AST nodes

5.1.2.12 cid_CVQualifiers

```
const ktc_childId_t cid_CVQualifiers
```

Constant to identify 'CVQualifiers' child of AST nodes

5.1.2.13 cid_Decl

```
const ktc_childId_t cid_Decl
```

Constant to identify 'Decl' child of AST nodes

5.1.2.14 cid_Declarator

```
const ktc_childId_t cid_Declarator
```

Constant to identify 'Declarator' child of AST nodes

5.1.2.15 cid_Declarators

```
const ktc_childId_t cid_Declarators
```

Constant to identify 'Declarators' child of AST nodes

5.1.2.16 cid_Decls

```
const ktc_childId_t cid_Decls
```

Constant to identify 'Decls' child of AST nodes

5.1.2.17 cid_DeclSpecs

```
const ktc_childId_t cid_DeclSpecs
```

Constant to identify 'DeclSpecs' child of AST nodes

5.1.2.18 cid_Default

```
const ktc_childId_t cid_Default
```

Constant to identify 'Default' child of AST nodes

5.1.2.19 cid_Designators

```
const ktc_childId_t cid_Designators
```

Constant to identify 'Designators' child of AST nodes

5.1.2.20 cid_Else

```
const ktc_childId_t cid_Else
```

Constant to identify 'Else' child of AST nodes

5.1.2.21 cid_Enumerators

```
const ktc_childId_t cid_Enumerators
```

Constant to identify 'Enumerators' child of AST nodes

5.1.2.22 cid_Exception

```
const ktc_childId_t cid_Exception
```

Constant to identify 'Exception' child of AST nodes

5.1.2.23 cid_Expr

```
const ktc_childId_t cid_Expr
```

Constant to identify 'Expr' child of AST nodes

5.1.2.24 cid_Func

```
const ktc_childId_t cid_Func
```

Constant to identify 'Func' child of AST nodes

5.1.2.25 cid_FuncBody

```
const ktc_childId_t cid_FuncBody
```

Constant to identify 'FuncBody' child of AST nodes

5.1.2.26 cid_Handler

```
const ktc_childId_t cid_Handler
```

Constant to identify 'Handler' child of AST nodes

5.1.2.27 cid_Handlers

```
const ktc_childId_t cid_Handlers
```

Constant to identify 'Handlers' child of AST nodes

5.1.2.28 cid_Index

```
const ktc_childId_t cid_Index
```

Constant to identify 'Index' child of AST nodes

5.1.2.29 cid_Init

```
const ktc_childId_t cid_Init
```

Constant to identify 'Init' child of AST nodes

5.1.2.30 cid_Initializer

```
const ktc_childId_t cid_Initializer
```

Constant to identify 'Initializer' child of AST nodes

5.1.2.31 cid_Inits

```
const ktc_childId_t cid_Inits
```

Constant to identify 'Inits' child of AST nodes

5.1.2.32 cid_Introducer

```
const ktc_childId_t cid_Introducer
```

Constant to identify 'Introducer' child of AST nodes

5.1.2.33 cid_KRParams

```
const ktc_childId_t cid_KRParams
```

Constant to identify 'KRParams' child of AST nodes

5.1.2.34 cid_Label

```
const ktc_childId_t cid_Label
```

Constant to identify 'Label' child of AST nodes

5.1.2.35 cid_LambdaCapture

```
const ktc_childId_t cid_LambdaCapture
```

Constant to identify 'LambdaCapture' child of AST nodes

5.1.2.36 cid_Left

```
const ktc_childId_t cid_Left
```

Constant to identify 'Left' child of AST nodes

5.1.2.37 cid_Literal

```
const ktc_childId_t cid_Literal
```

Constant to identify 'Literal' child of AST nodes

5.1.2.38 cid_Lower

```
const ktc_childId_t cid_Lower
```

Constant to identify 'Lower' child of AST nodes

5.1.2.39 cid_MemberDecl

```
const ktc_childId_t cid_MemberDecl
```

Constant to identify 'MemberDecl' child of AST nodes

5.1.2.40 cid_MemberDecls

```
const ktc_childId_t cid_MemberDecls
```

Constant to identify 'MemberDecls' child of AST nodes

5.1.2.41 cid_MemberInitializers

```
const ktc_childId_t cid_MemberInitializers
```

Constant to identify 'MemberInitializers' child of AST nodes

5.1.2.42 cid_Name

```
const ktc_childId_t cid_Name
```

Constant to identify 'Name' child of AST nodes

5.1.2.43 cid_NameSpec

```
const ktc_childId_t cid_NameSpec
```

Constant to identify 'NameSpec' child of AST nodes

5.1.2.44 cid_Next

```
const ktc_childId_t cid_Next
```

Constant to identify 'Next' child of AST nodes

5.1.2.45 cid_Params

```
const ktc_childId_t cid_Params
```

Constant to identify 'Params' child of AST nodes

5.1.2.46 cid_Placement

```
const ktc_childId_t cid_Placement
```

Constant to identify 'Placement' child of AST nodes

5.1.2.47 cid_PropertyFuncs

```
const ktc_childId_t cid_PropertyFuncs
```

Constant to identify 'PropertyFuncs' child of AST nodes

5.1.2.48 cid_Qualifier

```
const ktc_childId_t cid_Qualifier
```

Constant to identify 'Qualifier' child of AST nodes

5.1.2.49 cid_Right

```
const ktc_childId_t cid_Right
```

Constant to identify 'Right' child of AST nodes

5.1.2.50 cid_Size

```
const ktc_childId_t cid_Size
```

Constant to identify 'Size' child of AST nodes

5.1.2.51 cid_Stmt

```
const ktc_childId_t cid_Stmt
```

Constant to identify 'Stmt' child of AST nodes

5.1.2.52 cid_Stmts

```
const ktc_childId_t cid_Stmts
```

Constant to identify 'Stmts' child of AST nodes

5.1.2.53 cid_TemplateName

```
const ktc_childId_t cid_TemplateName
```

Constant to identify 'TemplateName' child of AST nodes

5.1.2.54 cid_TemplateParams

```
const ktc_childId_t cid_TemplateParams
```

Constant to identify 'TemplateParams' child of AST nodes

5.1.2.55 cid_Then

```
const ktc_childId_t cid_Then
```

Constant to identify 'Then' child of AST nodes

5.1.2.56 cid_Throw

```
const ktc_childId_t cid_Throw
```

Constant to identify 'Throw' child of AST nodes

5.1.2.57 cid_TrailingReturnType

```
const ktc_childId_t cid_TrailingReturnType
```

Constant to identify 'TrailingReturnType' child of AST nodes

5.1.2.58 cid_Type

```
const ktc_childId_t cid_Type
```

Constant to identify 'Type' child of AST nodes

5.1.2.59 cid_TypeIds

```
const ktc_childId_t cid_TypeIds
```

Constant to identify 'TypeIds' child of AST nodes

5.1.2.60 cid_Upper

```
const ktc_childId_t cid_Upper
```

Constant to identify 'Upper' child of AST nodes

5.1.2.61 tid_AccessSpecification

```
const ktc_treeType_t tid_AccessSpecification
```

Constant to identify 'AccessSpecification' AST node type

5.1.2.62 tid_AliasDecl

```
const ktc_treeType_t tid_AliasDecl
```

Constant to identify 'AliasDecl' AST node type

5.1.2.63 tid_AlignAsExpr

```
const ktc_treeType_t tid_AlignAsExpr
```

Constant to identify 'AlignAsExpr' AST node type

5.1.2.64 tid_AlignAsType

```
const ktc_treeType_t tid_AlignAsType
```

Constant to identify 'AlignAsType' AST node type

5.1.2.65 tid_AlignOfExpr

```
const ktc_treeType_t tid_AlignOfExpr
```

Constant to identify 'AlignOfExpr' AST node type

5.1.2.66 tid_Any

```
const ktc_treeType_t tid_Any
```

Constant to identify any AST node type

5.1.2.67 tid_AnyAttribute

```
const ktc_treeType_t tid_AnyAttribute
```

Constant to identify 'AnyAttribute' AST node type

5.1.2.68 tid_AnyCapture

```
const ktc_treeType_t tid_AnyCapture
```

Constant to identify 'AnyCapture' AST node type

5.1.2.69 tid_AnyDecl

```
const ktc_treeType_t tid_AnyDecl
```

Constant to identify 'AnyDecl' AST node type

5.1.2.70 tid_AnyDeclarator

```
const ktc_treeType_t tid_AnyDeclarator
```

Constant to identify 'AnyDeclarator' AST node type

5.1.2.71 tid_AnyDesignator

```
const ktc_treeType_t tid_AnyDesignator
```

Constant to identify 'AnyDesignator' AST node type

5.1.2.72 tid_AnyEnumerator

```
const ktc_treeType_t tid_AnyEnumerator
```

Constant to identify 'AnyEnumerator' AST node type

5.1.2.73 tid_AnyExpr

```
const ktc_treeType_t tid_AnyExpr
```

Constant to identify 'AnyExpr' AST node type

5.1.2.74 tid_AnyFuncBody

```
const ktc_treeType_t tid_AnyFuncBody
```

Constant to identify 'AnyFuncBody' AST node type

5.1.2.75 tid_AnyInitializer

```
const ktc_treeType_t tid_AnyInitializer
```

Constant to identify 'AnyInitializer' AST node type

5.1.2.76 tid_AnyLabel

```
const ktc_treeType_t tid_AnyLabel
```

Constant to identify 'AnyLabel' AST node type

5.1.2.77 tid_AnyMemberDecl

```
const ktc_treeType_t tid_AnyMemberDecl
```

Constant to identify 'AnyMemberDecl' AST node type

5.1.2.78 tid_AnyName

```
const ktc_treeType_t tid_AnyName
```

Constant to identify 'AnyName' AST node type

5.1.2.79 tid_AnyNameQualifier

```
const ktc_treeType_t tid_AnyNameQualifier
```

Constant to identify 'AnyNameQualifier' AST node type

5.1.2.80 tid_AnyNames

```
const ktc_treeType_t tid_AnyNames
```

Constant to identify 'AnyNames' AST node type

5.1.2.81 tid_AnyNameSpec

```
const ktc_treeType_t tid_AnyNameSpec
```

Constant to identify 'AnyNameSpec' AST node type

5.1.2.82 tid_AnyNonPtrDeclarator

```
const ktc_treeType_t tid_AnyNonPtrDeclarator
```

Constant to identify 'AnyNonPtrDeclarator' AST node type

5.1.2.83 tid_AnyParamName

```
const ktc_treeType_t tid_AnyParamName
```

Constant to identify 'AnyParamName' AST node type

5.1.2.84 tid_AnyPropertyFunc

```
const ktc_treeType_t tid_AnyPropertyFunc
```

Constant to identify 'AnyPropertyFunc' AST node type

5.1.2.85 tid_AnyPseudoDtor

```
const ktc_treeType_t tid_AnyPseudoDtor
```

Constant to identify 'AnyPseudoDtor' AST node type

5.1.2.86 tid_AnyStmt

```
const ktc_treeType_t tid_AnyStmt
```

Constant to identify 'AnyStmt' AST node type

5.1.2.87 tid_AnyTemplateArg

```
const ktc_treeType_t tid_AnyTemplateArg
```

Constant to identify 'AnyTemplateArg' AST node type

5.1.2.88 tid_AnyTypeName

```
const ktc_treeType_t tid_AnyTypeName
```

Constant to identify 'AnyTypeName' AST node type

5.1.2.89 tid_AnyTypeOf

```
const ktc_treeType_t tid_AnyTypeOf
```

Constant to identify 'AnyTypeOf' AST node type

5.1.2.90 tid_AnyTypeParam

```
const ktc_treeType_t tid_AnyTypeParam
```

Constant to identify 'AnyTypeParam' AST node type

5.1.2.91 tid_AnyUsing

```
const ktc_treeType_t tid_AnyUsing
```

Constant to identify 'AnyUsing' AST node type

5.1.2.92 tid_ArrayDeclarator

```
const ktc_treeType_t tid_ArrayDeclarator
```

Constant to identify 'ArrayDeclarator' AST node type

5.1.2.93 tid_AsmDef

```
const ktc_treeType_t tid_AsmDef
```

Constant to identify 'AsmDef' AST node type

5.1.2.94 tid_AsmStmt

```
const ktc_treeType_t tid_AsmStmt
```

Constant to identify 'AsmStmt' AST node type

5.1.2.95 tid_Attribute

```
const ktc_treeType_t tid_Attribute
```

Constant to identify 'Attribute' AST node type

5.1.2.96 tid_AttributedDeclarator

```
const ktc_treeType_t tid_AttributedDeclarator
```

Constant to identify 'AttributedDeclarator' AST node type

5.1.2.97 tid_AttributeDeclSpec

```
const ktc_treeType_t tid_AttributeDeclSpec
```

Constant to identify 'AttributeDeclSpec' AST node type

5.1.2.98 tid_Attributes

```
const ktc_treeType_t tid_Attributes
```

Constant to identify 'Attributes' AST node type

5.1.2.99 tid_AttributeSpec

```
const ktc_treeType_t tid_AttributeSpec
```

Constant to identify 'AttributeSpec' AST node type

5.1.2.100 tid_AttributeSpecs

```
const ktc_treeType_t tid_AttributeSpecs
```

Constant to identify 'AttributeSpecs' AST node type

5.1.2.101 tid_AttributeWithArgs

```
const ktc_treeType_t tid_AttributeWithArgs
```

Constant to identify 'AttributeWithArgs' AST node type

5.1.2.102 tid_AutoType

```
const ktc_treeType_t tid_AutoType
```

Constant to identify 'AutoType' AST node type

5.1.2.103 tid_BaseSpec

```
const ktc_treeType_t tid_BaseSpec
```

Constant to identify 'BaseSpec' AST node type

5.1.2.104 tid_BaseSpecs

```
const ktc_treeType_t tid_BaseSpecs
```

Constant to identify 'BaseSpecs' AST node type

5.1.2.105 tid_BinaryExpr

```
const ktc_treeType_t tid_BinaryExpr
```

Constant to identify 'BinaryExpr' AST node type

5.1.2.106 tid_BitFieldDeclarator

```
const ktc_treeType_t tid_BitFieldDeclarator
```

Constant to identify 'BitFieldDeclarator' AST node type

5.1.2.107 tid_BoolLiteralExpr

```
const ktc_treeType_t tid_BoolLiteralExpr
```

Constant to identify 'BoolLiteralExpr' AST node type

5.1.2.108 tid_BreakStmt

```
const ktc_treeType_t tid_BreakStmt
```

Constant to identify 'BreakStmt' AST node type

5.1.2.109 tid_BuiltinType

```
const ktc_treeType_t tid_BuiltinType
```

Constant to identify 'BuiltinType' AST node type

5.1.2.110 tid_CallExpr

```
const ktc_treeType_t tid_CallExpr
```

Constant to identify 'CallExpr' AST node type

5.1.2.111 tid_Capture

```
const ktc_treeType_t tid_Capture
```

Constant to identify 'Capture' AST node type

5.1.2.112 tid_CaptureDefault

```
const ktc_treeType_t tid_CaptureDefault
```

Constant to identify 'CaptureDefault' AST node type

5.1.2.113 tid_CaseLabel

```
const ktc_treeType_t tid_CaseLabel
```

Constant to identify 'CaseLabel' AST node type

5.1.2.114 tid_CaseRangeLabel

```
const ktc_treeType_t tid_CaseRangeLabel
```

Constant to identify 'CaseRangeLabel' AST node type

5.1.2.115 tid_CastExpr

```
const ktc_treeType_t tid_CastExpr
```

Constant to identify 'CastExpr' AST node type

5.1.2.116 tid_ClassType

```
const ktc_treeType_t tid_ClassType
```

Constant to identify 'ClassType' AST node type

5.1.2.117 tid_CompoundStmt

```
const ktc_treeType_t tid_CompoundStmt
```

Constant to identify 'CompoundStmt' AST node type

5.1.2.118 tid_ConditionalExpr

```
const ktc_treeType_t tid_ConditionalExpr
```

Constant to identify 'ConditionalExpr' AST node type

5.1.2.119 tid_ConstExpr

```
const ktc_treeType_t tid_ConstExpr
```

Constant to identify 'ConstExpr' AST node type

5.1.2.120 tid_ContinueStmt

```
const ktc_treeType_t tid_ContinueStmt
```

Constant to identify 'ContinueStmt' AST node type

5.1.2.121 tid_ConvFunc

```
const ktc_treeType_t tid_ConvFunc
```

Constant to identify 'ConvFunc' AST node type

5.1.2.122 tid_CopyInitializer

```
const ktc_treeType_t tid_CopyInitializer
```

Constant to identify 'CopyInitializer' AST node type

5.1.2.123 tid_CtorInitializer

```
const ktc_treeType_t tid_CtorInitializer
```

Constant to identify 'CtorInitializer' AST node type

5.1.2.124 tid_CVQualifier

```
const ktc_treeType_t tid_CVQualifier
```

Constant to identify 'CVQualifier' AST node type

5.1.2.125 tid_Decl

```
const ktc_treeType_t tid_Decl
```

Constant to identify 'Decl' AST node type

5.1.2.126 tid_DeclEllipsis

```
const ktc_treeType_t tid_DeclEllipsis
```

Constant to identify 'DeclEllipsis' AST node type

5.1.2.127 tid_DeclOrStmt

```
const ktc_treeType_t tid_DeclOrStmt
```

Constant to identify 'DeclOrStmt' AST node type

5.1.2.128 tid_DeclOrStmts

```
const ktc_treeType_t tid_DeclOrStmts
```

Constant to identify 'DeclOrStmts' AST node type

5.1.2.129 tid_DeclSpec

```
const ktc_treeType_t tid_DeclSpec
```

Constant to identify 'DeclSpec' AST node type

5.1.2.130 tid_DeclSpecs

```
const ktc_treeType_t tid_DeclSpecs
```

Constant to identify 'DeclSpecs' AST node type

5.1.2.131 tid_DefaultException

```
const ktc_treeType_t tid_DefaultException
```

Constant to identify 'DefaultException' AST node type

5.1.2.132 tid_DefaultLabel

```
const ktc_treeType_t tid_DefaultLabel
```

Constant to identify 'DefaultLabel' AST node type

5.1.2.133 tid_DeleteExpr

```
const ktc_treeType_t tid_DeleteExpr
```

Constant to identify 'DeleteExpr' AST node type

5.1.2.134 tid_DenyThrowSpec

```
const ktc_treeType_t tid_DenyThrowSpec
```

Constant to identify 'DenyThrowSpec' AST node type

5.1.2.135 tid_Designators

```
const ktc_treeType_t tid_Designators
```

Constant to identify 'Designators' AST node type

5.1.2.136 tid_DirectInitializer

```
const ktc_treeType_t tid_DirectInitializer
```

Constant to identify 'DirectInitializer' AST node type

5.1.2.137 tid_DoDeclStmt

```
const ktc_treeType_t tid_DoDeclStmt
```

Constant to identify 'DoDeclStmt' AST node type

5.1.2.138 tid_DoStmt

```
const ktc_treeType_t tid_DoStmt
```

Constant to identify 'DoStmt' AST node type

5.1.2.139 tid_Dtor

```
const ktc_treeType_t tid_Dtor
```

Constant to identify 'Dtor' AST node type

5.1.2.140 tid_Enumerator

```
const ktc_treeType_t tid_Enumerator
```

Constant to identify 'Enumerator' AST node type

5.1.2.141 tid_Enumerators

```
const ktc_treeType_t tid_Enumerators
```

Constant to identify 'Enumerators' AST node type

5.1.2.142 tid_EnumType

```
const ktc_treeType_t tid_EnumType
```

Constant to identify 'EnumType' AST node type

5.1.2.143 tid_ExceptHandler

```
const ktc_treeType_t tid_ExceptHandler
```

Constant to identify 'ExceptHandler' AST node type

5.1.2.144 tid_Exception

```
const ktc_treeType_t tid_Exception
```

Constant to identify 'Exception' AST node type

5.1.2.145 tid_ExceptionSpec

```
const ktc_treeType_t tid_ExceptionSpec
```

Constant to identify 'ExceptionSpec' AST node type

5.1.2.146 tid_ExplicitInstantiation

```
const ktc_treeType_t tid_ExplicitInstantiation
```

Constant to identify 'ExplicitInstantiation' AST node type

5.1.2.147 tid_ExprArg

```
const ktc_treeType_t tid_ExprArg
```

Constant to identify 'ExprArg' AST node type

5.1.2.148 tid_Exprs

```
const ktc_treeType_t tid_Exprs
```

Constant to identify 'Exprs' AST node type

5.1.2.149 tid_ExprStmt

```
const ktc_treeType_t tid_ExprStmt
```

Constant to identify 'ExprStmt' AST node type

5.1.2.150 tid_ExprTypeIdExpr

```
const ktc_treeType_t tid_ExprTypeIdExpr
```

Constant to identify 'ExprTypeIdExpr' AST node type

5.1.2.151 tid_FieldDesignator

```
const ktc_treeType_t tid_FieldDesignator
```

Constant to identify 'FieldDesignator' AST node type

5.1.2.152 tid_FinallyHandler

```
const ktc_treeType_t tid_FinallyHandler
```

Constant to identify 'FinallyHandler' AST node type

5.1.2.153 tid_ForEachStmt

```
const ktc_treeType_t tid_ForEachStmt
```

Constant to identify 'ForEachStmt' AST node type

5.1.2.154 tid_ForRangeStmt

```
const ktc_treeType_t tid_ForRangeStmt
```

Constant to identify 'ForRangeStmt' AST node type

5.1.2.155 tid_ForStmt

```
const ktc_treeType_t tid_ForStmt
```

Constant to identify 'ForStmt' AST node type

5.1.2.156 tid_FuncBody

```
const ktc_treeType_t tid_FuncBody
```

Constant to identify 'FuncBody' AST node type

5.1.2.157 tid_FuncDeclarator

```
const ktc_treeType_t tid_FuncDeclarator
```

Constant to identify 'FuncDeclarator' AST node type

5.1.2.158 tid_FuncDef

```
const ktc_treeType_t tid_FuncDef
```

Constant to identify 'FuncDef' AST node type

5.1.2.159 tid_FuncSpec

```
const ktc_treeType_t tid_FuncSpec
```

Constant to identify 'FuncSpec' AST node type

5.1.2.160 tid_FuncTryBlock

```
const ktc_treeType_t tid_FuncTryBlock
```

Constant to identify 'FuncTryBlock' AST node type

5.1.2.161 tid_GenericAttribute

```
const ktc_treeType_t tid_GenericAttribute
```

Constant to identify 'GenericAttribute' AST node type

5.1.2.162 tid_GlobalScope

```
const ktc_treeType_t tid_GlobalScope
```

Constant to identify 'GlobalScope' AST node type

5.1.2.163 tid_GotoStmt

```
const ktc_treeType_t tid_GotoStmt
```

Constant to identify 'GotoStmt' AST node type

5.1.2.164 tid_Handler

```
const ktc_treeType_t tid_Handler
```

Constant to identify 'Handler' AST node type

5.1.2.165 tid_Handlers

```
const ktc_treeType_t tid_Handlers
```

Constant to identify 'Handlers' AST node type

5.1.2.166 tid_IdExpr

```
const ktc_treeType_t tid_IdExpr
```

Constant to identify 'IdExpr' AST node type

5.1.2.167 tid_IfDeclStmt

```
const ktc_treeType_t tid_IfDeclStmt
```

Constant to identify 'IfDeclStmt' AST node type

5.1.2.168 tid_IfStmt

```
const ktc_treeType_t tid_IfStmt
```

Constant to identify 'IfStmt' AST node type

5.1.2.169 tid_IndexDesignator

```
const ktc_treeType_t tid_IndexDesignator
```

Constant to identify 'IndexDesignator' AST node type

5.1.2.170 tid_IndexExpr

```
const ktc_treeType_t tid_IndexExpr
```

Constant to identify 'IndexExpr' AST node type

5.1.2.171 tid_InitClause

```
const ktc_treeType_t tid_InitClause
```

Constant to identify 'InitClause' AST node type

5.1.2.172 tid_InitializedDeclarator

```
const ktc_treeType_t tid_InitializedDeclarator
```

Constant to identify 'InitializedDeclarator' AST node type

5.1.2.173 tid_InitializerExpr

```
const ktc_treeType_t tid_InitializerExpr
```

Constant to identify 'InitializerExpr' AST node type

5.1.2.174 tid_Initializers

```
const ktc_treeType_t tid_Initializers
```

Constant to identify 'Initializers' AST node type

5.1.2.175 tid_KRFuncDeclarator

```
const ktc_treeType_t tid_KRFuncDeclarator
```

Constant to identify 'KRFuncDeclarator' AST node type

5.1.2.176 tid_Label

```
const ktc_treeType_t tid_Label
```

Constant to identify 'Label' AST node type

5.1.2.177 tid_LabeledStmt

```
const ktc_treeType_t tid_LabeledStmt
```

Constant to identify 'LabeledStmt' AST node type

5.1.2.178 tid_LambdaDeclarator

```
const ktc_treeType_t tid_LambdaDeclarator
```

Constant to identify 'LambdaDeclarator' AST node type

5.1.2.179 tid_LambdaExpr

```
const ktc_treeType_t tid_LambdaExpr
```

Constant to identify 'LambdaExpr' AST node type

5.1.2.180 tid_LambdaIntroducer

```
const ktc_treeType_t tid_LambdaIntroducer
```

Constant to identify 'LambdaIntroducer' AST node type

5.1.2.181 tid_LeaveStmt

```
const ktc_treeType_t tid_LeaveStmt
```

Constant to identify 'LeaveStmt' AST node type

5.1.2.182 tid_LinkageSpec

```
const ktc_treeType_t tid_LinkageSpec
```

Constant to identify 'LinkageSpec' AST node type

5.1.2.183 tid_LiteralExpr

```
const ktc_treeType_t tid_LiteralExpr
```

Constant to identify 'LiteralExpr' AST node type

5.1.2.184 tid_MaybeCtorInitializer

```
const ktc_treeType_t tid_MaybeCtorInitializer
```

Constant to identify 'MaybeCtorInitializer' AST node type

5.1.2.185 tid_MaybeDeclarator

```
const ktc_treeType_t tid_MaybeDeclarator
```

Constant to identify 'MaybeDeclarator' AST node type

5.1.2.186 tid_MaybeException

```
const ktc_treeType_t tid_MaybeException
```

Constant to identify 'MaybeException' AST node type

5.1.2.187 tid_MaybeExceptionSpec

```
const ktc_treeType_t tid_MaybeExceptionSpec
```

Constant to identify 'MaybeExceptionSpec' AST node type

5.1.2.188 tid_MaybeLambdaDeclarator

```
const ktc_treeType_t tid_MaybeLambdaDeclarator
```

Constant to identify 'MaybeLambdaDeclarator' AST node type

5.1.2.189 tid_MaybeNewInitializer

```
const ktc_treeType_t tid_MaybeNewInitializer
```

Constant to identify 'MaybeNewInitializer' AST node type

5.1.2.190 tid_MaybeTypeId

```
const ktc_treeType_t tid_MaybeTypeId
```

Constant to identify 'MaybeTypeId' AST node type

5.1.2.191 tid_MemberDecl

```
const ktc_treeType_t tid_MemberDecl
```

Constant to identify 'MemberDecl' AST node type

5.1.2.192 tid_MemberDecls

```
const ktc_treeType_t tid_MemberDecls
```

Constant to identify 'MemberDecls' AST node type

5.1.2.193 tid_MemberDesignator

```
const ktc_treeType_t tid_MemberDesignator
```

Constant to identify 'MemberDesignator' AST node type

5.1.2.194 tid_MemberExpr

```
const ktc_treeType_t tid_MemberExpr
```

Constant to identify 'MemberExpr' AST node type

5.1.2.195 tid_MemberFunc

```
const ktc_treeType_t tid_MemberFunc
```

Constant to identify 'MemberFunc' AST node type

5.1.2.196 tid_MemberInitializer

```
const ktc_treeType_t tid_MemberInitializer
```

Constant to identify 'MemberInitializer' AST node type

5.1.2.197 tid_MemberInitializers

```
const ktc_treeType_t tid_MemberInitializers
```

Constant to identify 'MemberInitializers' AST node type

5.1.2.198 tid_MemberTemplate

```
const ktc_treeType_t tid_MemberTemplate
```

Constant to identify 'MemberTemplate' AST node type

5.1.2.199 tid_MemberUsingDecl

```
const ktc_treeType_t tid_MemberUsingDecl
```

Constant to identify 'MemberUsingDecl' AST node type

5.1.2.200 tid_Name

```
const ktc_treeType_t tid_Name
```

Constant to identify 'Name' AST node type

5.1.2.201 tid_NameDeclarator

```
const ktc_treeType_t tid_NameDeclarator
```

Constant to identify 'NameDeclarator' AST node type

5.1.2.202 tid_NamespaceAlias

```
const ktc_treeType_t tid_NamespaceAlias
```

Constant to identify 'NamespaceAlias' AST node type

5.1.2.203 tid_NamespaceDecl

```
const ktc_treeType_t tid_NamespaceDecl
```

Constant to identify 'NamespaceDecl' AST node type

5.1.2.204 tid_NameSpec

```
const ktc_treeType_t tid_NameSpec
```

Constant to identify 'NameSpec' AST node type

5.1.2.205 tid_NewExpr

```
const ktc_treeType_t tid_NewExpr
```

Constant to identify 'NewExpr' AST node type

5.1.2.206 tid_NewInitializer

```
const ktc_treeType_t tid_NewInitializer
```

Constant to identify 'NewInitializer' AST node type

5.1.2.207 tid_NoAttribute

```
const ktc_treeType_t tid_NoAttribute
```

Constant to identify 'NoAttribute' AST node type

5.1.2.208 tid_NoAttributeSpec

```
const ktc_treeType_t tid_NoAttributeSpec
```

Constant to identify 'NoAttributeSpec' AST node type

5.1.2.209 tid_NoBaseSpec

```
const ktc_treeType_t tid_NoBaseSpec
```

Constant to identify 'NoBaseSpec' AST node type

5.1.2.210 tid_NoCapture

```
const ktc_treeType_t tid_NoCapture
```

Constant to identify 'NoCapture' AST node type

5.1.2.211 tid_NoCtorInitializer

```
const ktc_treeType_t tid_NoCtorInitializer
```

Constant to identify 'NoCtorInitializer' AST node type

5.1.2.212 tid_Node

```
const ktc_treeType_t tid_Node
```

Constant to identify 'Node' AST node type

5.1.2.213 tid_NoDeclarator

```
const ktc_treeType_t tid_NoDeclarator
```

Constant to identify 'NoDeclarator' AST node type

5.1.2.214 tid_NoDeclOrStmt

```
const ktc_treeType_t tid_NoDeclOrStmt
```

Constant to identify 'NoDeclOrStmt' AST node type

5.1.2.215 tid_NoDeclSpec

```
const ktc_treeType_t tid_NoDeclSpec
```

Constant to identify 'NoDeclSpec' AST node type

5.1.2.216 tid_NoDesignator

```
const ktc_treeType_t tid_NoDesignator
```

Constant to identify 'NoDesignator' AST node type

5.1.2.217 tid_NoEnumerator

```
const ktc_treeType_t tid_NoEnumerator
```

Constant to identify 'NoEnumerator' AST node type

5.1.2.218 tid_NoException

```
const ktc_treeType_t tid_NoException
```

Constant to identify 'NoException' AST node type

5.1.2.219 tid_NoExceptionSpec

```
const ktc_treeType_t tid_NoExceptionSpec
```

Constant to identify 'NoExceptionSpec' AST node type

5.1.2.220 tid_NoExpr

```
const ktc_treeType_t tid_NoExpr
```

Constant to identify 'NoExpr' AST node type

5.1.2.221 tid_NoHandler

```
const ktc_treeType_t tid_NoHandler
```

Constant to identify 'NoHandler' AST node type

5.1.2.222 tid_NoInitializer

```
const ktc_treeType_t tid_NoInitializer
```

Constant to identify 'NoInitializer' AST node type

5.1.2.223 tid_NoLambdaDeclarator

```
const ktc_treeType_t tid_NoLambdaDeclarator
```

Constant to identify 'NoLambdaDeclarator' AST node type

5.1.2.224 tid_NoMemberDecl

```
const ktc_treeType_t tid_NoMemberDecl
```

Constant to identify 'NoMemberDecl' AST node type

5.1.2.225 tid_NoMemberInitializer

```
const ktc_treeType_t tid_NoMemberInitializer
```

Constant to identify 'NoMemberInitializer' AST node type

5.1.2.226 tid_NoName

```
const ktc_treeType_t tid_NoName
```

Constant to identify 'NoName' AST node type

5.1.2.227 tid_NoNameQualifier

```
const ktc_treeType_t tid_NoNameQualifier
```

Constant to identify 'NoNameQualifier' AST node type

5.1.2.228 tid_NoNewInitializer

```
const ktc_treeType_t tid_NoNewInitializer
```

Constant to identify 'NoNewInitializer' AST node type

5.1.2.229 tid_NoParamName

```
const ktc_treeType_t tid_NoParamName
```

Constant to identify 'NoParamName' AST node type

5.1.2.230 tid_NoPropertyFunc

```
const ktc_treeType_t tid_NoPropertyFunc
```

Constant to identify 'NoPropertyFunc' AST node type

5.1.2.231 tid_NoTemplateArg

```
const ktc_treeType_t tid_NoTemplateArg
```

Constant to identify 'NoTemplateArg' AST node type

5.1.2.232 tid_NoTemplateParam

```
const ktc_treeType_t tid_NoTemplateParam
```

Constant to identify 'NoTemplateParam' AST node type

5.1.2.233 tid_NoTypeId

```
const ktc_treeType_t tid_NoTypeId
```

Constant to identify 'NoTypeId' AST node type

5.1.2.234 tid_NullptrLiteralExpr

```
const ktc_treeType_t tid_NullptrLiteralExpr
```

Constant to identify 'NullptrLiteralExpr' AST node type

5.1.2.235 tid_OpFunc

```
const ktc_treeType_t tid_OpFunc
```

Constant to identify 'OpFunc' AST node type

5.1.2.236 tid_Param

```
const ktc_treeType_t tid_Param
```

Constant to identify 'Param' AST node type

5.1.2.237 tid_ParamName

```
const ktc_treeType_t tid_ParamName
```

Constant to identify 'ParamName' AST node type

5.1.2.238 tid_ParamNames

```
const ktc_treeType_t tid_ParamNames
```

Constant to identify 'ParamNames' AST node type

5.1.2.239 tid_ParensDeclarator

```
const ktc_treeType_t tid_ParensDeclarator
```

Constant to identify 'ParensDeclarator' AST node type

5.1.2.240 tid_ParensExpr

```
const ktc_treeType_t tid_ParensExpr
```

Constant to identify 'ParensExpr' AST node type

5.1.2.241 tid_PromisedFuncBody

```
const ktc_treeType_t tid_PromisedFuncBody
```

Constant to identify 'PromisedFuncBody' AST node type

5.1.2.242 tid_PromisedMemberDecl

```
const ktc_treeType_t tid_PromisedMemberDecl
```

Constant to identify 'PromisedMemberDecl' AST node type

5.1.2.243 tid_PropertyAttribute

```
const ktc_treeType_t tid_PropertyAttribute
```

Constant to identify 'PropertyAttribute' AST node type

5.1.2.244 tid_PropertyFuncs

```
const ktc_treeType_t tid_PropertyFuncs
```

Constant to identify 'PropertyFuncs' AST node type

5.1.2.245 tid_PropertyGetFunc

```
const ktc_treeType_t tid_PropertyGetFunc
```

Constant to identify 'PropertyGetFunc' AST node type

5.1.2.246 tid_PropertyPutFunc

```
const ktc_treeType_t tid_PropertyPutFunc
```

Constant to identify 'PropertyPutFunc' AST node type

5.1.2.247 tid_PseudoDtor

```
const ktc_treeType_t tid_PseudoDtor
```

Constant to identify 'PseudoDtor' AST node type

5.1.2.248 tid_PtrDeclarator

```
const ktc_treeType_t tid_PtrDeclarator
```

Constant to identify 'PtrDeclarator' AST node type

5.1.2.249 tid_QualifiedName

```
const ktc_treeType_t tid_QualifiedName
```

Constant to identify 'QualifiedName' AST node type

5.1.2.250 tid_QualifiedPseudoDtor

```
const ktc_treeType_t tid_QualifiedPseudoDtor
```

Constant to identify 'QualifiedPseudoDtor' AST node type

5.1.2.251 tid_RangeDesignator

```
const ktc_treeType_t tid_RangeDesignator
```

Constant to identify 'RangeDesignator' AST node type

5.1.2.252 tid_ReservedTypeSpec

```
const ktc_treeType_t tid_ReservedTypeSpec
```

Constant to identify 'ReservedTypeSpec' AST node type

5.1.2.253 tid_ReturnStmt

```
const ktc_treeType_t tid_ReturnStmt
```

Constant to identify 'ReturnStmt' AST node type

5.1.2.254 tid_SizeOfExpr

```
const ktc_treeType_t tid_SizeOfExpr
```

Constant to identify 'SizeOfExpr' AST node type

5.1.2.255 tid_SpecialCastExpr

```
const ktc_treeType_t tid_SpecialCastExpr
```

Constant to identify 'SpecialCastExpr' AST node type

5.1.2.256 tid_StaticAssertDecl

```
const ktc_treeType_t tid_StaticAssertDecl
```

Constant to identify 'StaticAssertDecl' AST node type

5.1.2.257 tid_StmtExpr

```
const ktc_treeType_t tid_StmtExpr
```

Constant to identify 'StmtExpr' AST node type

5.1.2.258 tid_StorageClass

```
const ktc_treeType_t tid_StorageClass
```

Constant to identify 'StorageClass' AST node type

5.1.2.259 tid_StringLiteralExpr

```
const ktc_treeType_t tid_StringLiteralExpr
```

Constant to identify 'StringLiteralExpr' AST node type

5.1.2.260 tid_SuffixFunc

```
const ktc_treeType_t tid_SuffixFunc
```

Constant to identify 'SuffixFunc' AST node type

5.1.2.261 tid_SuperScope

```
const ktc_treeType_t tid_SuperScope
```

Constant to identify 'SuperScope' AST node type

5.1.2.262 tid_SwitchDeclStmt

```
const ktc_treeType_t tid_SwitchDeclStmt
```

Constant to identify 'SwitchDeclStmt' AST node type

5.1.2.263 tid_SwitchStmt

```
const ktc_treeType_t tid_SwitchStmt
```

Constant to identify 'SwitchStmt' AST node type

5.1.2.264 tid_TemplateArgs

```
const ktc_treeType_t tid_TemplateArgs
```

Constant to identify 'TemplateArgs' AST node type

5.1.2.265 tid_TemplateDecl

```
const ktc_treeType_t tid_TemplateDecl
```

Constant to identify 'TemplateDecl' AST node type

5.1.2.266 tid_TemplateName

```
const ktc_treeType_t tid_TemplateName
```

Constant to identify 'TemplateName' AST node type

5.1.2.267 tid_TemplateParam

```
const ktc_treeType_t tid_TemplateParam
```

Constant to identify 'TemplateParam' AST node type

5.1.2.268 tid_TemplateParams

```
const ktc_treeType_t tid_TemplateParams
```

Constant to identify 'TemplateParams' AST node type

5.1.2.269 tid_TemplateSpec

```
const ktc_treeType_t tid_TemplateSpec
```

Constant to identify 'TemplateSpec' AST node type

5.1.2.270 tid_TemplateTypeArg

```
const ktc_treeType_t tid_TemplateTypeArg
```

Constant to identify 'TemplateTypeArg' AST node type

5.1.2.271 tid_TemplateTypeParam

```
const ktc_treeType_t tid_TemplateTypeParam
```

Constant to identify 'TemplateTypeParam' AST node type

5.1.2.272 tid_ThisExpr

```
const ktc_treeType_t tid_ThisExpr
```

Constant to identify 'ThisExpr' AST node type

5.1.2.273 tid_ThrowExpr

```
const ktc_treeType_t tid_ThrowExpr
```

Constant to identify 'ThrowExpr' AST node type

5.1.2.274 tid_TranslationUnit

```
const ktc_treeType_t tid_TranslationUnit
```

Constant to identify 'TranslationUnit' AST node type

5.1.2.275 tid_TruncatedInitClause

```
const ktc_treeType_t tid_TruncatedInitClause
```

Constant to identify 'TruncatedInitClause' AST node type

5.1.2.276 tid_TryExceptStmt

```
const ktc_treeType_t tid_TryExceptStmt
```

Constant to identify 'TryExceptStmt' AST node type

5.1.2.277 tid_TryFinallyStmt

```
const ktc_treeType_t tid_TryFinallyStmt
```

Constant to identify 'TryFinallyStmt' AST node type

5.1.2.278 tid_TryStmt

```
const ktc_treeType_t tid_TryStmt
```

Constant to identify 'TryStmt' AST node type

5.1.2.279 tid_TypeAdjective

```
const ktc_treeType_t tid_TypeAdjective
```

Constant to identify 'TypeAdjective' AST node type

5.1.2.280 tid_TypeArg

```
const ktc_treeType_t tid_TypeArg
```

Constant to identify 'TypeArg' AST node type

5.1.2.281 tid_TypeConvExpr

```
const ktc_treeType_t tid_TypeConvExpr
```

Constant to identify 'TypeConvExpr' AST node type

5.1.2.282 tid_TypeId

```
const ktc_treeType_t tid_TypeId
```

Constant to identify 'TypeId' AST node type

5.1.2.283 tid_TypeName

```
const ktc_treeType_t tid_TypeName
```

Constant to identify 'TypeName' AST node type

5.1.2.284 tid_TypeOfExpr

```
const ktc_treeType_t tid_TypeOfExpr
```

Constant to identify 'TypeOfExpr' AST node type

5.1.2.285 tid_TypeOfSpec

```
const ktc_treeType_t tid_TypeOfSpec
```

Constant to identify 'TypeOfSpec' AST node type

5.1.2.286 tid_TypeOfType

```
const ktc_treeType_t tid_TypeOfType
```

Constant to identify 'TypeOfType' AST node type

5.1.2.287 tid_TypeParam

```
const ktc_treeType_t tid_TypeParam
```

Constant to identify 'TypeParam' AST node type

5.1.2.288 tid_TypeTypeIdExpr

```
const ktc_treeType_t tid_TypeTypeIdExpr
```

Constant to identify 'TypeTypeIdExpr' AST node type

5.1.2.289 tid_UnaryExpr

```
const ktc_treeType_t tid_UnaryExpr
```

Constant to identify 'UnaryExpr' AST node type

5.1.2.290 tid_UnparsedDecl

```
const ktc_treeType_t tid_UnparsedDecl
```

Constant to identify 'UnparsedDecl' AST node type

5.1.2.291 tid_UnparsedDeclarator

```
const ktc_treeType_t tid_UnparsedDeclarator
```

Constant to identify 'UnparsedDeclarator' AST node type

5.1.2.292 tid_UnparsedDeclSpec

```
const ktc_treeType_t tid_UnparsedDeclSpec
```

Constant to identify 'UnparsedDeclSpec' AST node type

5.1.2.293 tid_UnparsedEnumerator

```
const ktc_treeType_t tid_UnparsedEnumerator
```

Constant to identify 'UnparsedEnumerator' AST node type

5.1.2.294 tid_UnparsedException

```
const ktc_treeType_t tid_UnparsedException
```

Constant to identify 'UnparsedException' AST node type

5.1.2.295 tid_UnparsedExpr

```
const ktc_treeType_t tid_UnparsedExpr
```

Constant to identify 'UnparsedExpr' AST node type

5.1.2.296 tid_UnparsedInitializer

```
const ktc_treeType_t tid_UnparsedInitializer
```

Constant to identify 'UnparsedInitializer' AST node type

5.1.2.297 tid_UnparsedLabel

```
const ktc_treeType_t tid_UnparsedLabel
```

Constant to identify 'UnparsedLabel' AST node type

5.1.2.298 tid_UnparsedMemberDecl

```
const ktc_treeType_t tid_UnparsedMemberDecl
```

Constant to identify 'UnparsedMemberDecl' AST node type

5.1.2.299 tid_UnparsedName

```
const ktc_treeType_t tid_UnparsedName
```

Constant to identify 'UnparsedName' AST node type

5.1.2.300 tid_UnparsedNameQualifier

```
const ktc_treeType_t tid_UnparsedNameQualifier
```

Constant to identify 'UnparsedNameQualifier' AST node type

5.1.2.301 tid_UnparsedParamName

```
const ktc_treeType_t tid_UnparsedParamName
```

Constant to identify 'UnparsedParamName' AST node type

5.1.2.302 tid_UnparsedPropertyFunc

```
const ktc_treeType_t tid_UnparsedPropertyFunc
```

Constant to identify 'UnparsedPropertyFunc' AST node type

5.1.2.303 tid_UnparsedStmt

```
const ktc_treeType_t tid_UnparsedStmt
```

Constant to identify 'UnparsedStmt' AST node type

5.1.2.304 tid_UnqualifiedName

```
const ktc_treeType_t tid_UnqualifiedName
```

Constant to identify 'UnqualifiedName' AST node type

5.1.2.305 tid_UserLiteralExpr

```
const ktc_treeType_t tid_UserLiteralExpr
```

Constant to identify 'UserLiteralExpr' AST node type

5.1.2.306 tid_UserStringLiteralExpr

```
const ktc_treeType_t tid_UserStringLiteralExpr
```

Constant to identify 'UserStringLiteralExpr' AST node type

5.1.2.307 tid_UsingDecl

```
const ktc_treeType_t tid_UsingDecl
```

Constant to identify 'UsingDecl' AST node type

5.1.2.308 tid_UsingDirective

```
const ktc_treeType_t tid_UsingDirective
```

Constant to identify 'UsingDirective' AST node type

5.1.2.309 tid_WhileDeclStmt

```
const ktc_treeType_t tid_WhileDeclStmt
```

Constant to identify 'WhileDeclStmt' AST node type

5.1.2.310 tid_WhileStmt

```
const ktc_treeType_t tid_WhileStmt
```

Constant to identify 'WhileStmt' AST node type

5.2 ktcAPI.h File Reference

```
#include "ktcMainAPI.h"
#include "gen-ktcAPI.h"
```

5.3 ktcMainAPI.h File Reference

```
#include "kwapi.h"
#include <stdint.h>
```

Macros

- #define **KTC_API_VERSION_MAJOR** 1
- #define **KTC_API_VERSION_MINOR** 9
- #define **KTC_API_VERSION_PATCHLEVEL** 0
- #define **KTC_CUSTOM_TYPES**
- #define **ktc_require**(t, ttype) if (! **ktc_isTreeType**((t),(ttype))) { return 0; }

Typedefs

- typedef union CTree_Node ∗ **ktc_tree_t**
- typedef union rs_element ∗ **ktc_semanticInfo_t**
- typedef union rs_element ∗ **ktc_languageType_t**
- typedef struct ktc_string_t_ ∗ **ktc_string_t**
- typedef int64_t **ktc_long_long_t**
- typedef int **ktc_treeType_t**
- typedef int **ktc_childId_t**
- typedef int(∗ **ktc_treeHook_t**) (**ktc_tree_t**)
- typedef void(∗ **ktc_eventHook_t**) (void)
- typedef void ∗ **ktc_position_t**
- typedef **ktc_position_t** **ktc_position**
- typedef void ∗ **ktc_autofix_t**
- typedef void ∗ **ktc_message_t**
- typedef void ∗ **ktc_message**

Functions

- int **ktc_isTreeType** (**ktc_tree_t** t, **ktc_treeType_t** ttype)
- const char ∗ **ktc_treeType_getName** (**ktc_tree_t** ttype)
- **ktc_tree_t** **ktc_proceed** (**ktc_tree_t** t, **ktc_childId_t** child_id)
- void **ktc_forAllSubtreeNodes** (**ktc_tree_t** t, int(∗callback)(**ktc_tree_t**, void ∗), void ∗data)
- void **ktc_sema_forAllSubtreeNodes** (**ktc_semanticInfo_t** si, int(∗callback)(**ktc_tree_t**, void ∗), void ∗data)
- int **ktc_isMacroExpansion** (**ktc_tree_t** t)
- int **ktc_isMacroExpansion2** (**ktc_tree_t** t)
- int **ktc_nodeStackTop** (void)
- **ktc_tree_t** **ktc_nodeStackGet** (int n)
- void **ktc_registerTreeHook** (int tree_event, **ktc_treeType_t** tt, **ktc_treeHook_t** p_hook)
- void **ktc_registerStartTraverseHook** (**ktc_eventHook_t** p_hook)
- void **ktc_registerSaveContextHook** (**ktc_eventHook_t** p_hook)
- void **ktc_registerRestoreContextHook** (**ktc_eventHook_t** p_hook)
- void **ktc_registerStopTraverseHook** (**ktc_eventHook_t** p_hook)
- void **ktc_unregisterTreeHook** (int tree_event, **ktc_treeType_t** tt, **ktc_treeHook_t** p_hook)
- **ktc_semanticInfo_t** **ktc_getSemanticInfo** (**ktc_tree_t** t)
- **ktc_semanticInfo_t** **ktc_getCalledFunction** (**ktc_tree_t** t)

- **ktc_semanticInfo_t ktc_getAssociatedScope** (**ktc_tree_t** t)
- **ktc_semanticInfo_t ktc_sema_getOverridenMethod** (**ktc_tree_t** t)
- int **ktc_sema_isSameFunctions** (**ktc_semanticInfo_t** si1, **ktc_semanticInfo_t** si2)
- int **ktc_sema_isSameVariables** (**ktc_semanticInfo_t** si1, **ktc_semanticInfo_t** si2)
- int **ktc_sema_haveSameSignature** (**ktc_semanticInfo_t** si1, **ktc_semanticInfo_t** si2)
- int **ktc_sema_haveSameFunctionType** (**ktc_semanticInfo_t** si1, **ktc_semanticInfo_t** si2)
- int **ktc_sema_isSameClass** (**ktc_semanticInfo_t** si1, **ktc_semanticInfo_t** si2)
- int **ktc_sema_isSameEnum** (**ktc_semanticInfo_t** si1, **ktc_semanticInfo_t** si2)
- int **ktc_sema_isSameScope** (**ktc_semanticInfo_t** si1, **ktc_semanticInfo_t** si2)
- int **ktc_sema_isNone** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isBitfield** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isScope** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isClass** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isUnion** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isTemplate** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isEnum** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isEnumConstant** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isType** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isVariable** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isPointer** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isArray** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isReference** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isFunctionType** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isFunction** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isBuiltin** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isSpecialization** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isInstantiation** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isInstantiatedFunction** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isIntegerValue** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isObjectValue** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isFunctionTemplateSet** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isFunctionTemplate** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isNamespaceAlias** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isUsingDirective** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isUsingDeclaration** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isUsing** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isNamespace** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isPOD** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isFriend** (**ktc_semanticInfo_t** si)
- int **ktc_sema_hasMethods** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getClassTag** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isAnonymous** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isTypeParameter** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isVirtual** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isPrivate** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isProtected** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isPublic** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isPureVirtual** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isConstructor** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isDestructor** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isOperatorFunction** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isLocal** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isGlobal** (**ktc_semanticInfo_t** si)
- **ktc_tree_t ktc_sema_getVariableInitializer** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getVariableType** (**ktc_semanticInfo_t** si)

- **ktc_semanticInfo_t ktc_sema_getVariableValue** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getNumberOfArguments** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getFormalArgument** (**ktc_semanticInfo_t** si, int n)
- **ktc_semanticInfo_t ktc_sema_getReturnType** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getFunctionType** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getBuiltinCode** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getCVQualifiers** (**ktc_semanticInfo_t** si)
- int **ktc_sema_isImmutable** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getPointedType** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getArrayElementType** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getArraySize** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getReferencedType** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getDefinedType** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_skipTypedefs** (**ktc_semanticInfo_t** si, int ∗cvq)
- const char ∗ **ktc_sema_getIdentifier** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getIdentifierNo** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getNumber** (**ktc_semanticInfo_t** si)
- const char ∗ **ktc_sema_getTypedefedName** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_stripCVQ** (**ktc_semanticInfo_t** si)
- char ∗ **ktc_sema_getQualifiedName** (**ktc_semanticInfo_t** si)
- char ∗ **ktc_sema_getTypeName** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getScope** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getNumberOfBaseInfo** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getBaseInfo** (**ktc_semanticInfo_t** si, int i)
- int **ktc_sema_isBaseVirtual** (**ktc_semanticInfo_t** si, int i)
- int **ktc_sema_isBasePublic** (**ktc_semanticInfo_t** si, int i)
- int **ktc_sema_isBaseProtected** (**ktc_semanticInfo_t** si, int i)
- int **ktc_sema_isBasePrivate** (**ktc_semanticInfo_t** si, int i)
- int **ktc_sema_isConstMethod** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t ktc_sema_getFirstByName** (**ktc_semanticInfo_t** scope, const char ∗name)
- **kw_array_t** ∗ **ktc_sema_getAllByName** (**ktc_semanticInfo_t** scope, const char ∗name)
- **ktc_semanticInfo_t ktc_sema_findFirstByName** (**ktc_semanticInfo_t** scope, const char ∗name)
- **ktc_semanticInfo_t ktc_sema_getGlobalScope** ()
- **kw_array_t** ∗ **ktc_sema_findAllByName** (**ktc_semanticInfo_t** scope, const char ∗name)
- void **ktc_sema_forAllClassDeclarations** (**ktc_semanticInfo_t** class_info, void(∗callback)(**ktc_semanticInfo_t**))
- void **ktc_sema_forAllScopeDeclarations** (**ktc_semanticInfo_t** scope_info, void(∗callback)(**ktc_semanticInfo_t**))
- int **ktc_sema_functionOverloadsFunction** (**ktc_semanticInfo_t** func1, **ktc_semanticInfo_t** func2)
- int **ktc_sema_RelatedClasses** (**ktc_semanticInfo_t** class1, **ktc_semanticInfo_t** class2)
- **ktc_languageType_t ktc_getLanguageType** (**ktc_tree_t** node)
- int **ktc_languageTypeSize** (**ktc_languageType_t** lang_type)
- int **ktc_languageTypeIsPointer** (**ktc_languageType_t** type)
- **ktc_languageType_t ktc_getPointedType** (**ktc_languageType_t** ptr_type)
- int **ktc_languageTypeIsBuiltin** (**ktc_languageType_t** type, int builtin_code)
- int **ktc_languageTypeSignedness** (**ktc_languageType_t** ctypeinfo)
- int **ktc_languageTypeEffectiveSignedness** (**ktc_languageType_t** ctypeinfo)
- int **ktc_isCallTo** (**ktc_tree_t** node, const char ∗fn_name)
- **ktc_tree_t ktc_getCallArgument** (**ktc_tree_t** node, int n_arg)
- int **ktc_getNumberOfCallArguments** (**ktc_tree_t** node)
- int **ktc_isCharLiteral** (**ktc_tree_t** node)
- **ktc_long_long_t ktc_getIntegerValue** (**ktc_tree_t** t, int ∗error_flag)
- **ktc_long_long_t ktc_sema_getIntegerValue** (**ktc_semanticInfo_t** val, int ∗error_flag)
- const char ∗ **ktc_sema_getObjectName** (**ktc_semanticInfo_t** si)
- int **ktc_sema_getIntValueType** (**ktc_semanticInfo_t** si)

- int **ktc_isNullPointerConstant** (**ktc_tree_t** t)
- int **ktc_isNullMacro** (**ktc_tree_t** t)
- **ktc_tree_t ktc_getSizeofArgument** (**ktc_tree_t** t)
- int **ktc_assembleStringConstant** (**ktc_tree_t** exprString, char ∗∗pbuf)
- int **ktc_isWideString** (**ktc_tree_t** exprString)
- int **ktc_isUTF16String** (**ktc_tree_t** exprString)
- int **ktc_isUTF32String** (**ktc_tree_t** exprString)
- const char ∗ **ktc_getNoIdent** (void)
- const char ∗ **ktc_getIdentifier** (**ktc_tree_t** t)
- int **ktc_getIdentifierNo** (**ktc_tree_t** t)
- char ∗ **ktc_getTokens** (**ktc_tree_t** t)
- int **ktc_is_NoToken** (**ktc_tree_t** t)
- int **ktc_is_Token** (**ktc_tree_t** t)
- char ∗ **ktc_getStringConstantValue** (**ktc_tree_t** t)
- **ktc_tree_t ktc_skipBrackets** (**ktc_tree_t** t)
- **ktc_tree_t ktc_getNameDeclarator** (**ktc_tree_t**)
- int **ktc_getOperation** (**ktc_tree_t** t)
- int **ktc_isOperationOverloaded** (**ktc_tree_t** t)
- int **ktc_getStorageClass** (**ktc_tree_t** t)
- int **ktc_getPointerOperator** (**ktc_tree_t** t)
- int **ktc_getTypeQualifiers** (**ktc_tree_t** t)
- int **ktc_getClassTag** (**ktc_tree_t** t)
- int **ktc_getBuiltinType** (**ktc_tree_t** t)
- int **ktc_getBuiltinTypeSize** (const char ∗type_name)
- int **ktc_getPointerSize** ()
- int **ktc_getCastSpecifier** (**ktc_tree_t** t)
- int **ktc_getFunctionSpecifier** (**ktc_tree_t** t)
- int **ktc_compareSubtrees** (**ktc_tree_t** t1, **ktc_tree_t** t2)
- **ktc_position_t ktc_position_new** (int line, int col, const char ∗fname)
- **ktc_position_t ktc_position_copy** (**ktc_position_t** pos)
- void **ktc_position_delete** (**ktc_position_t** pos)
- **ktc_position_t ktc_getStartPosition** (**ktc_tree_t** t)
- **ktc_position_t ktc_getEndPosition** (**ktc_tree_t** t)
- int **ktc_position_getLine** (**ktc_position_t** pos)
- void **ktc_position_setLine** (**ktc_position_t** pos, int line)
- int **ktc_position_getColumn** (**ktc_position_t** pos)
- char ∗ **ktc_position_getFileName** (**ktc_position_t** pos)
- **ktc_autofix_t ktc_autofix_new** ()
- void **ktc_autofix_delete** (**ktc_autofix_t** a)
- void **ktc_autofix_addSegment** (**ktc_autofix_t** a, const char ∗repl, const char ∗file, int line1, int col1, int line2, int col2)
- int **ktc_error_isEnabled** (const char ∗error_id)
- const char ∗ **ktc_error_getConfigurationParameter** (const char ∗error_id, const char ∗param_name)
- int **ktc_getFrontendLanguage** (void)
- int **ktc_getFrontendDialect** (void)
- const char ∗ **ktc_getOutputFileName** (void)
- int **ktc_hasBuiltinWideChar** (void)
- **ktc_message_t ktc_message_new** (const char ∗error_id)
- void **ktc_message_setPosition** (**ktc_message_t** msg, **ktc_position_t** pos)
- void **ktc_message_setRecommendationFactor** (**ktc_message_t** msg, const char ∗factor, int value)
- void **ktc_message_addAttribute** (**ktc_message_t** msg, const char ∗attr_string)
- void **ktc_message_addAnchorAttribute** (**ktc_message_t** msg, const char ∗attr_string)
- void **ktc_message_addTraceBySemanticsInfo** (**ktc_message_t** msg, **ktc_semanticInfo_t** sema)
- void **ktc_message_addEvent** (**ktc_message_t** msg, **ktc_position_t** pos, const char ∗str)
- void ∗ **ktc_event_new** (**ktc_position_t** pos, const char ∗str)

- void **ktc_event_setParameter** (void ∗event, const char ∗name, const char ∗value)
- void **ktc_message_addEventEx** (void ∗msg, void ∗event)
- void **ktc_message_render** (**ktc_message_t** msg)
- void **ktc_message_render_wi_autofix** (**ktc_message_t** msg, **ktc_autofix_t** a)
- void **ktc_message_delete** (**ktc_message_t** msg)
- void **ktc_message_setFunction** (void ∗function)
- void **ktc_message_unsetFunction** ()
- int **ktc_isStatic** (**ktc_tree_t** node)
- int **ktc_sema_isStatic** (**ktc_semanticInfo_t** info)
- int **ktc_isNameIncluded** (const char ∗fname)
- int **ktc_isIncluded** (**ktc_tree_t** node)
- int **ktc_sema_isIncluded** (**ktc_semanticInfo_t** si)
- void **ktc_free** (void ∗ptr)
- int **ktc_isConstructor** (**ktc_tree_t** node)
- int **ktc_sema_getFieldNumber** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t** **ktc_sema_getFieldType** (**ktc_semanticInfo_t** si, int fieldpos)
- **ktc_position_t** **ktc_sema_getPosition** (**ktc_semanticInfo_t** si)
- **ktc_tree_t** **ktc_sema_getAST** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t** **ktc_sema_getPrimaryTemplate** (**ktc_semanticInfo_t** si)
- **kw_array_t** ∗ **ktc_sema_getInstantiationParameters** (**ktc_semanticInfo_t** si)
- **kw_array_t** ∗ **ktc_sema_getEnumerators** (**ktc_semanticInfo_t** si)
- int **ktc_isDeclaration** (**ktc_tree_t** info)
- int **ktc_isDefinition** (**ktc_tree_t** info)
- int **ktc_sema_isFuncDef** (**ktc_semanticInfo_t** info)
- **ktc_semanticInfo_t** **ktc_sema_getInstantiatedClass** (**ktc_semanticInfo_t** si)
- **ktc_semanticInfo_t** **ktc_sema_getFunctionFromTemplate** (**ktc_semanticInfo_t** info)
- **kw_array_t** ∗ **ktc_sema_getFunctionTemplateSpecializations** (**ktc_semanticInfo_t** info)
- **kw_array_t** ∗ **ktc_sema_getFunctionTemplateInstantiations** (**ktc_semanticInfo_t** info)
- **ktc_semanticInfo_t** **ktc_sema_getInstantiationOrigin** (**ktc_semanticInfo_t** info)
- short **ktc_sema_checkBitField** (**ktc_semanticInfo_t** info)
- int **ktc_sema_isFunctionPointer** (**ktc_semanticInfo_t** info)
- **ktc_semanticInfo_t** **ktc_sema_getFunctionPointerType** (**ktc_semanticInfo_t** info)
- int **ktc_sema_isInline** (**ktc_semanticInfo_t** info)
- **ktc_semanticInfo_t** **ktc_sema_getEnumUnderlyingTypeOrInt** (**ktc_semanticInfo_t** info)
- **ktc_string_t** **ktc_string_new** (const char ∗s)
- void **ktc_string_delete** (**ktc_string_t** ks)
- const char ∗ **ktc_string_get_cstring** (**ktc_string_t** ks)

Variables

- int **KTC_TREE_EVENT_ON_ENTER**
- int **KTC_TREE_EVENT_ON_NEXT**
- int **KTC_TREE_EVENT_ON_LEAVE**
- const char ∗ **ktc_constructorName**
- const char ∗ **ktc_destructorName**
- int **KTC_TYPESIGNEDNESS_NONE**
- int **KTC_TYPESIGNEDNESS_DEFAULT**
- int **KTC_TYPESIGNEDNESS_SIGNED**
- int **KTC_TYPESIGNEDNESS_UNSIGNED**
- const int **KTC_LANGUAGE_C**
- const int **KTC_LANGUAGE_CXX**
- const int **KTC_DIALECT_STD**
- const int **KTC_DIALECT_GNU**
- const int **KTC_DIALECT_MS**
- const int **KTC_DIALECT_ARM**
- const int **KTC_DIALECT_GHS**
- const int **KTC_DIALECT_SUN**
- const int **KTC_DIALECT_TI**

5.3.1 Macro Definition Documentation

5.3.1.1 KTC_API_VERSION_MAJOR

```
#define KTC_API_VERSION_MAJOR 1
```

5.3.1.2 KTC_API_VERSION_MINOR

```
#define KTC_API_VERSION_MINOR 9
```

5.3.1.3 KTC_API_VERSION_PATCHLEVEL

```
#define KTC_API_VERSION_PATCHLEVEL 0
```

5.3.1.4 KTC_CUSTOM_TYPES

```
#define KTC_CUSTOM_TYPES
```

5.3.1.5 ktc_require

```
#define ktc_require(
            t,
            ttype ) if (!  ktc_isTreeType((t),(ttype))) { return 0; }
```

Require that node has a particular kind and return from handler with 0 (not applicable) otherwise

Parameters

t	subtree root to check
ttype	required tree type

5.3.2 Typedef Documentation

5.3.2.1 ktc_autofix_t

```
typedef void* ktc_autofix_t
```

Opaque data type that describes autofix (file, line, and column)

5.3.2.2 ktc_childId_t

```
typedef int ktc_childId_t
```

Type definition for child enumeration constants

5.3.2.3 ktc_eventHook_t

```
typedef void(* ktc_eventHook_t) (void)
```

5.3.2.4 ktc_languageType_t

```
typedef union rs_element* ktc_languageType_t
```

Opaque type representing C/C++ language type description Can safely casted to **ktc_semanticInfo_t** (p. 262) pointer type

Deprecated Use **ktc_semanticInfo_t** (p. 262) type

5.3.2.5 ktc_long_long_t

```
typedef int64_t ktc_long_long_t
```

Type definition for the largest available integer type

5.3.2.6 ktc_message

```
typedef void* ktc_message
```

Alias for ktc_message_t - for compatibility with 1.0 - 1.1 API versions

5.3.2.7 ktc_message_t

```
typedef void* ktc_message_t
```

Opaque type for warning/error message data structure

5.3.2.8 ktc_position

```
typedef ktc_position_t ktc_position
```

Alias for ktc_position_t - for compatibility with 1.0 - 1.1 API versions

5.3.2.9 ktc_position_t

```
typedef void* ktc_position_t
```

Opaque data type that describes position (file, line, and column)

5.3.2.10 ktc_semanticInfo_t

```
typedef union rs_element* ktc_semanticInfo_t
```

Opaque type for accessing semantic description of symbols. Can be safely casted to **ktc_languageType_t** (p. 261) pointer type described below

5.3.2.11 ktc_string_t

```
typedef struct ktc_string_t_* ktc_string_t
```

Opaque type for representing strings returned by KAST custom functions

5.3.2.12 ktc_tree_t

```
typedef union CTree_Node* ktc_tree_t
```

Opaque type representing an Abstract Syntax subtree, starting with one node

5.3.2.13 ktc_treeHook_t

```
typedef int(* ktc_treeHook_t) ( ktc_tree_t)
```

5.3.2.14 ktc_treeType_t

```
typedef int ktc_treeType_t
```

Type definition for tree type

5.3.3 Function Documentation

5.3.3.1 ktc_assembleStringConstant()

```
int ktc_assembleStringConstant (
            ktc_tree_t exprString,
            char ** pbuf )
```

Return buffer with a string constant assembled from a list of adjacent string literals. Buffer is allocated on heap, so calling procedure must take care to 'free' this buffer when it is no longer needed.

Remarks

Please not that string pointer returned via pointer in 'pbuf' is *not* a C zero-terminated string. String literal may contain zero bytes.

Parameters

exprString	tree root for the first literal in the list of adjacent string literals
pbuf	address of a pointer that will point to a newly allocated buffer with a string

Returns

length of the whole string buffer. If exprString is not of type tid_StringLiteralExpr, 0 is returned.

5.3.3.2 ktc_autofix_addSegment()

```
void ktc_autofix_addSegment (
            ktc_autofix_t a,
            const char * repl,
            const char * file,
            int line1,
            int col1,
            int line2,
            int col2 )
```

5.3.3.3 ktc_autofix_delete()

```
void ktc_autofix_delete (
            ktc_autofix_t a )
```

5.3.3.4 ktc_autofix_new()

```
ktc_autofix_t ktc_autofix_new ( )
```

Create newly allocated position with line, col and file name

Parameters

t	subtree to get start position of

Returns

newly allocated position, that can be deallocated with **ktc_autofix_delete()** (p. 264)

5.3.3.5 ktc_compareSubtrees()

```
int ktc_compareSubtrees (
            ktc_tree_t t1,
            ktc_tree_t t2 )
```

Compare two subtrees for structural equivalence. Subtrees are structurally equivalent if kinds of their root nodes are the same and all their children subtrees (excluding Next links) are structurally equivalent.

Parameters

t1	first subtree to compare
t2	second subtree to compare

Returns

1 if subtrees are structurally equivalent, 0 otherwise

5.3.3.6 ktc_error_getConfigurationParameter()

```
const char* ktc_error_getConfigurationParameter (
            const char * error_id,
            const char * param_name )
```

Get configuration parameter for an error

Parameters

param_name	name of the configuration parameter
error_id	identifier of an error (the same id as in <error> tag in configuration file)

Returns

parameter value or "" if error was not configured or there is no such parameter

5.3.3.7 ktc_error_isEnabled()

```
int ktc_error_isEnabled (
            const char * error_id )
```

Deprecated Use kwapi_cfgparam_errorIsEnabled

5.3.3.8 ktc_event_new()

```
void* ktc_event_new (
            ktc_position_t pos,
            const char * str )
```

5.3.3.9 ktc_event_setParameter()

```
void ktc_event_setParameter (
            void * event,
            const char * name,
            const char * value )
```

5.3.3.10 ktc_forAllSubtreeNodes()

```
void ktc_forAllSubtreeNodes (
            ktc_tree_t t,
            int(*)( ktc_tree_t, void *) callback,
            void * data )
```

From given subtree root, apply the given callback to all subtree nodes

Parameters

t	subtree root to start with
callback	callback to be applied The provided callback should return: 1 if no further traversal is required at all, 2 if no further children traversal is required, but Next links should be explored, and 0 otherwise. Node stack related routines should not be used from a callback

5.3.3.11 ktc_free()

```
void ktc_free (
            void * ptr )
```

Deallocate memory

Parameters

ptr	is pointer to memoty allocated from ktc_ functions

5.3.3.12 ktc_getAssociatedScope()

```
ktc_semanticInfo_t ktc_getAssociatedScope (
            ktc_tree_t t )
```

Get semantic description about a scope associated with a language construct described by node Applicable to nodes of the following kinds: NamespaceDecl, CompoundStmt, IfDeclStmt, SwitchDeclStmt, WhileDeclStmt, Do↩ DeclStmt, ForStmt

Returns

pointer to semantic information or 0 if no corresponding information available

5.3.3.13 ktc_getBuiltinType()

```
int ktc_getBuiltinType (
            ktc_tree_t t )
```

Get builtin type code from the tid_BuiltinType subtree. This function is not applicable to other node types, and so only basic builtin types can be returned. Note that the returned type is not necessary the actual type of a declaration Use **ktc_sema_getBuiltinCode** (p. 293) function to get the complete information

Parameters

t	expression subtree, must be of tree type tid_BuiltinType

Returns

operation code (see **Numerical codes of C/C++ built-in types** (p. 34)) or KTC_BUILTINTYPE_NONE if preconditions are not met

5.3.3.14 ktc_getBuiltinTypeSize()

```
int ktc_getBuiltinTypeSize (
            const char * type_name )
```

Get size (in bytes) of a builtin type by the type's name.

Parameters

type_name	name of builtin type

Returns

type size in bytes or -1 if the given type is unknown

5.3.3.15 ktc_getCallArgument()

```
ktc_tree_t ktc_getCallArgument (
            ktc_tree_t node,
           int n_arg )
```

Get n-th argument of a call, starting from 1

Parameters

node	call expression node
n_arg	number of an argument

Returns

a subtree for an n-th argument or 0 if 'n' is out of range or 'node' is not a call expression

5.3.3.16 ktc_getCalledFunction()

```
ktc_semanticInfo_t ktc_getCalledFunction (
            ktc_tree_t t )
```

Get semantic description of a function called in a language construct described by node Applicable to nodes of the following kinds: CallExpr, TypeConvExpr, IndexExpr, CastExpr

Returns

pointer to semantic information or 0 if no corresponding information available

5.3.3.17 ktc_getCastSpecifier()

```
int ktc_getCastSpecifier (
            ktc_tree_t t )
```

Get cast specifier code (for C++ -style cast expressions) from the tid_SpecialCastExpr subtree. This function is not applicable to other node types,

Parameters

t	expression subtree, must be of tree type tid_SpecialCastExpr

Returns

cast specifier code (see **Numerical codes for identifying different C++ -style cast expressions** (p. 39)) or -1 if t belongs to any other type.

5.3.3.18 ktc_getClassTag()

```
int ktc_getClassTag (
            ktc_tree_t t )
```

Get class tag code from the tid_ClassType subtree.

Parameters

t	expression subtree, must be of tree type tid_ClassType

Returns

operation code (see **Numerical codes to differentiate struct/class/union declarations** (p. 33)) or KTC_↩CLASSTAG_NONE if preconditions are not met

5.3.3.19 ktc_getEndPosition()

```
ktc_position_t ktc_getEndPosition (
            ktc_tree_t t )
```

Get end position of the code fragment that is translated to a particular subtree

Parameters

t	subtree to get end position of

5.3.3.20 ktc_getFrontendDialect()

```
int ktc_getFrontendDialect (
            void )
```

Returns

dialect of the current compilation (KTC_DIALECT_STD / KTC_DIALECT_GNU / KTC_DIALECT_MS / KTC↩_DIALECT_ARM / KTC_DIALECT_GHS / KTC_DIALECT_SUN / KTC_DIALECT_TI

5.3.3.21 ktc_getFrontendLanguage()

```
int ktc_getFrontendLanguage (
            void  )
```

Returns

programming language of the current compilation (KTC_LANGUAGE_CXX or KTC_LANGUAGE_C).

5.3.3.22 ktc_getFunctionSpecifier()

```
int ktc_getFunctionSpecifier (
            ktc_tree_t t )
```

Get function specifier code (for C++ member functions) from the tid_FunctionSpec subtree.

Parameters

t	expression subtree

Returns

function specifier code (see **Numerical codes for identifying inline/virtual/explicit/friend member function specifiers** (p. 38)), KTC_FUNCSPECIFIER_NONE is returned when no specifier is provided in the source or preconditions are not met.

5.3.3.23 ktc_getIdentifier()

```
const char* ktc_getIdentifier (
            ktc_tree_t t )
```

Get identifier string from tree nodes of subtypes: tid_IdExpr, tid_Name, tt_namespaceDef, tid_NamespaceAlias, tid_GotoStmt, tid_TypeName, tid_Enumerator, tid_NameDeclarator, tid_ParamName, tid_FieldDesignator, tid_↩MemberDesignator, tid_Label, tid_ExprEnum, tid_NameSpec, tid_Dtor

5.3.3.24 ktc_getIdentifierNo()

```
int ktc_getIdentifierNo (
            ktc_tree_t t )
```

5.3.3.25 ktc_getIntegerValue()

```
ktc_long_long_t ktc_getIntegerValue (
            ktc_tree_t t,
            int * error_flag )
```

Get integer value of constant of one of integer types or bool type (in C++)

Parameters

t	constant subtree
error_flag	set to non-zero value if it is impossible to convert constant into integer value

Returns

integer value of a constant, 0 is returned for 'false', 1 is returned for 'true'

5.3.3.26 ktc_getLanguageType()

```
ktc_languageType_t ktc_getLanguageType (
            ktc_tree_t node )
```

Get C/C++ type corresponding to subtree

Returns

NULL if type information is not available, pointer to type information otherwise

5.3.3.27 ktc_getNameDeclarator()

```
ktc_tree_t ktc_getNameDeclarator (
            ktc_tree_t  )
```

Get innermost declarator from a declarator sequence For example, in the declaration 'int ∗a[];', (after explicit breakdown it is 'int ∗((a)[]))') the outermost declarator is '∗a[]', then follows 'a[]', the innermost is 'a'.

5.3.3.28 ktc_getNoIdent()

```
const char* ktc_getNoIdent (
            void  )
```

DEPRECATED!

5.3.3.29 ktc_getNumberOfCallArguments()

```
int ktc_getNumberOfCallArguments (
            ktc_tree_t node )
```

Get number of call arguments

Parameters

node	call expression node

Returns

number of call arguments or (-1) if 'node' is not a call expression

5.3.3.30 ktc_getOperation()

```
int ktc_getOperation (
            ktc_tree_t t )
```

Get operation code from the Unary, Binary, Ternary or Filed expression subtree, from OperatorFunction subtree or from Operation subtree.

Parameters

t	expression subtree, must be of the following tree types: tid_UnaryExpr, tid_BinaryExpr, tid_MemberExpr, tid_OpFunc

Returns

operation code (see **Numerical codes of operations** (p. 22)) or KTC_OPCODE_NONE if preconditions are not met

5.3.3.31 ktc_getOutputFileName()

```
const char* ktc_getOutputFileName (
            void )
```

Returns

path to the output Klocwork object file

5.3.3.32 ktc_getPointedType()

```
ktc_languageType_t ktc_getPointedType (
            ktc_languageType_t ptr_type )
```

Get type pointed by given type

Returns

pointed type if ptr_type is a pointer type, 0 otherwise.

Deprecated Use **ktc_sema_getPointedType** (p. 301)

5.3.3.33 ktc_getPointerOperator()

```
int ktc_getPointerOperator (
            ktc_tree_t t )
```

Get pointer operator kind from tid_Ptr subtree.

Parameters

t	expression subtree, must be of tree type tid_Ptr

Returns

pointer operatior code code (see **Numerical codes for identifying declarator ptr types** (p. 40)) or KTC_P↩OINTEROPERATOR_NONE if preconditions are not met

5.3.3.34 ktc_getPointerSize()

```
int ktc_getPointerSize ( )
```

Get size (in bytes) of a pointer.

Returns

size of a pointer in bytes

5.3.3.35 ktc_getSemanticInfo()

```
ktc_semanticInfo_t ktc_getSemanticInfo (
            ktc_tree_t t )
```

Get semantic description of language element described by node Semantic information can be retrieved only for nodes of the following kinds: NamespaceAlias, UsingDecl1, UsingDecl2, Enumr, Designator1, Designator2, Qualifier1, BasicIdentifier, QualifiedIdent, NewTypeId, NoTypeId, TypeId, ExprField, ExprIdent, ExprTypeid, DirDeclr1, Declr, ClassDeclr, TemplateArg2, TemplateArg3, TypeName, TypeOf2

Returns

pointer to semantic information or 0 if no semantic information available

5.3.3.36 ktc_getSizeofArgument()

```
ktc_tree_t ktc_getSizeofArgument (
            ktc_tree_t t )
```

Return argument of 'sizeof' regardless of the fact whether its argument is a type or an expression

5.3.3.37 ktc_getStartPosition()

```
ktc_position_t ktc_getStartPosition (
            ktc_tree_t t )
```

Get start position of the code fragment that is translated to a particular subtree

Parameters

t	subtree to get start position of

5.3.3.38 ktc_getStorageClass()

```
int ktc_getStorageClass (
            ktc_tree_t t )
```

Get storage class code from the tid_StorageClass subtree.

Parameters

t	expression subtree, must be of tree type tid_StorageClass

Returns

operation code (see **Numerical codes of declaration storage class specifiers** (p. 30)) or KTC_STORAG↩ ECLASS_NONE if preconditions are not met

5.3.3.39 ktc_getStringConstantValue()

```
char* ktc_getStringConstantValue (
            ktc_tree_t t )
```

Get string from StringLiteralExpr

Returns

newly allocated string, that can be deallocated with **ktc_free()** (p. 266)

5.3.3.40 ktc_getTokens()

```
char* ktc_getTokens (
            ktc_tree_t t )
```

Get identifier string from tree using Tokens

Returns

newly allocated string, that can be deallocated with **ktc_free()** (p. 266)

5.3.3.41 ktc_getTypeQualifiers()

```
int ktc_getTypeQualifiers (
            ktc_tree_t t )
```

Get type qualifier flags from the tid_CVQualifier subtree.

Parameters

t	expression subtree, must be of tree type tid_CVQualifier

Returns

bit-or'ed superposition of type qualifier flags (see **Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values** (p. 32)) or KTC_CVQUALIFIER_NONE if preconditions are not met

5.3.3.42 ktc_hasBuiltinWideChar()

```
int ktc_hasBuiltinWideChar (
            void  )
```

Check if wchar_t is a built-in type

5.3.3.43 ktc_is_NoToken()

```
int ktc_is_NoToken (
            ktc_tree_t t )
```

Checks that root of subtree has 'NoToken' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'NoToken', 0 otherwise

5.3.3.44 ktc_is_Token()

```
int ktc_is_Token (
            ktc_tree_t t )
```

Checks that root of subtree has 'Token' AST node type.

Parameters

t	subtree root to test

Returns

1 if 't' belongs to the type 'Token', 0 otherwise

5.3.3.45 ktc_isCallTo()

```
int ktc_isCallTo (
            ktc_tree_t node,
            const char * fn_name )
```

Check that subtree is a representation of a call to a function with particular name.

Parameters

node	subtree root ot check
fn_name	function name

Returns

1 if subtree is a call of function with name 'fn_name', 0 otherwise

5.3.3.46 ktc_isCharLiteral()

```
int ktc_isCharLiteral (
            ktc_tree_t node )
```

Checks if the node represents a char literal

Parameters

node	the node to test

Returns

1 if the node represents a char literal, 0 otherwise

5.3.3.47 ktc_isConstructor()

```
int ktc_isConstructor (
            ktc_tree_t node )
```

Check that given node corresponds to a class constructor

Returns

1 if true; 0 if false

5.3.3.48 ktc_isDeclaration()

```
int ktc_isDeclaration (
            ktc_tree_t info )
```

Check that given node corresponds to a class or enum declaration

Returns

1 if true; 0 if false

5.3.3.49 ktc_isDefinition()

```
int ktc_isDefinition (
            ktc_tree_t info )
```

Check that given node corresponds to a class or enum definition

Returns

1 if true; 0 if false

5.3.3.50 ktc_isIncluded()

```
int ktc_isIncluded (
            ktc_tree_t node )
```

Check that given node is included via header file

Returns

1 if is; 0 otherwise

5.3.3.51 ktc_isMacroExpansion()

```
int ktc_isMacroExpansion (
            ktc_tree_t t )
```

Checks that subtree comes from a macro expansion.

Parameters

t	subtree root to test

Returns

1 if 't' comes from a macro expansion, 0 otherwise

5.3.3.52 ktc_isMacroExpansion2()

```
int ktc_isMacroExpansion2 (
            ktc_tree_t t )
```

- Checks that one of the subtrees come from a macro expansion.

Parameters

t	subtree root to test

- **Returns**
- 1 if 't' comes from a macro expansion, 0 otherwise

5.3.3.53 ktc_isNameIncluded()

```
int ktc_isNameIncluded (
            const char * fname )
```

Check that given file name is a name of a header file

Returns

1 if is; 0 otherwise

5.3.3.54 ktc_isNullMacro()

```
int ktc_isNullMacro (
            ktc_tree_t t )
```

Validate if the given tree represents a "NULL Macro".

5.3.3.55 ktc_isNullPointerConstant()

```
int ktc_isNullPointerConstant (
            ktc_tree_t t )
```

Validate if the given tree represents a "null pointer constant". According to the C standard section 6.3.2.3, a null pointer constant is "an integer constant expression with the value 0, or such an expression cast to type void ∗."

Here are some examples of accepted null pointer constants: 0 (void ∗) 0 (((0))) (((1 - 1))) (1 ^ 1)

Parameters

t	the tree

Returns

1 if the tree represents a null pointer constant, 0 otherwise.

5.3.3.56 ktc_isOperationOverloaded()

```
int ktc_isOperationOverloaded (
            ktc_tree_t t )
```

Check if operation is overloaded

Parameters

t	expression subtree, must be of the following tree types: tid_UnaryExpr, tid_BinaryExpr, tid_MemberExpr

5.3.3.57 ktc_isStatic()

```
int ktc_isStatic (
            ktc_tree_t node )
```

Check that given node corresponds to a static declaration

Returns

1 if true, 0 if false

5.3.3.58 ktc_isTreeType()

```
int ktc_isTreeType (
            ktc_tree_t t,
            ktc_treeType_t ttype )
```

Checks that root of subtree has particular node type. Node type is a generalization of node kind. Node kind is an instance of one of node types. Node type, in turn, may be a subtype of another node type

Parameters

t	subtree root to test
ttype	tree type identifier

Returns

1 if 't' belongs to the type 'ttype', 0 otherwise

5.3.3.59 ktc_isUTF16String()

```
int ktc_isUTF16String (
            ktc_tree_t exprString )
```

Return 1 if the given string literal is a utf-16 character string, 0 otherwise

Parameters

exprString	tree root for the first literal in the list of adjacent string literals

Returns

1 if the given string literal is a utf-16 character string, 0 otherwise

5.3.3.60 ktc_isUTF32String()

```
int ktc_isUTF32String (
            ktc_tree_t exprString )
```

Return 1 if the given string literal is a utf-32 character string, 0 otherwise

Parameters

exprString	tree root for the first literal in the list of adjacent string literals

Returns

1 if the given string literal is a utf-32 character string, 0 otherwise

5.3.3.61 ktc_isWideString()

```
int ktc_isWideString (
            ktc_tree_t exprString )
```

Return 1 if the given string literal is a wide character string, 0 otherwise

Parameters

exprString	tree root for the first literal in the list of adjacent string literals

Returns

1 if the given string literal is a wide character string, 0 otherwise

5.3.3.62 ktc_languageTypeEffectiveSignedness()

```
int ktc_languageTypeEffectiveSignedness (
            ktc_languageType_t ctypeinfo )
```

Get effective 'signedness' of a type

Returns

'signedness' value (see **Attribute values for signedness of types** (p. 13))

5.3.3.63 ktc_languageTypeIsBuiltin()

```
int ktc_languageTypeIsBuiltin (
            ktc_languageType_t type,
           int builtin_code )
```

Deprecated Use **ktc_sema_getBuiltinCode** (p. 293) instead

5.3.3.64 ktc_languageTypeIsPointer()

```
int ktc_languageTypeIsPointer (
            ktc_languageType_t type )
```

Check that C/C++ type is actually a pointer type

Returns

1 if type is a pointer type, 0 otherwise.

Deprecated Use **ktc_sema_isPointer** (p. 312)

5.3.3.65 ktc_languageTypeSignedness()

```
int ktc_languageTypeSignedness (
            ktc_languageType_t ctypeinfo )
```

Get 'signedness' of a type

Returns

'signedness' value (see **Attribute values for signedness of types** (p. 13))

5.3.3.66 ktc_languageTypeSize()

```
int ktc_languageTypeSize (
            ktc_languageType_t lang_type )
```

Get size taken by value of the type (as in sizeof)

Returns

size taken by a value of a given type

Remarks

size calculation routines do not respect alignment, so size of structure is counted as a sum of its members' sizes. Structure containing one 'int' and one 'char' occupies 8 bits on 32-bit platforms with 4-byte alignment and 6 bytes on 32-bit platforms with 2-byte alignment. However, this routine will return 5.

5.3.3.67 ktc_message_addAnchorAttribute()

```
void ktc_message_addAnchorAttribute (
            ktc_message_t msg,
            const char * attr_string )
```

5.3.3.68 ktc_message_addAttribute()

```
void ktc_message_addAttribute (
            ktc_message_t msg,
            const char * attr_string )
```

Add arbitrary attribute to the message, they may be referred by {n} in the format string. 'n' is an number of an attribute (starting from 0)

Parameters

msg	message to add attribute to
attr_string	attribute value

5.3.3.69 ktc_message_addEvent()

```
void ktc_message_addEvent (
            ktc_message_t msg,
            ktc_position_t pos,
           const char * str )
```

5.3.3.70 ktc_message_addEventEx()

```
void ktc_message_addEventEx (
           void * msg,
           void * event )
```

5.3.3.71 ktc_message_addTraceBySemanticsInfo()

```
void ktc_message_addTraceBySemanticsInfo (
            ktc_message_t msg,
            ktc_semanticInfo_t sema )
```

5.3.3.72 ktc_message_delete()

```
void ktc_message_delete (
            ktc_message_t msg )
```

Free memory occupied by internal message data structures

5.3.3.73 ktc_message_new()

```
ktc_message_t ktc_message_new (
           const char * error_id )
```

Allocate memory for an error report

Parameters

error↩ _id	identifier of an error (the same id as in <error> tag in configuration file)

Returns

Allocated structure for keeping message details

5.3.3.74 ktc_message_render()

```
void ktc_message_render (
            ktc_message_t msg )
```

Put message into the object file

5.3.3.75 ktc_message_render_wi_autofix()

```
void ktc_message_render_wi_autofix (
            ktc_message_t msg,
            ktc_autofix_t a )
```

Put message into the object file

5.3.3.76 ktc_message_setFunction()

```
void ktc_message_setFunction (
            void * function )
```

Set current function

5.3.3.77 ktc_message_setPosition()

```
void ktc_message_setPosition (
            ktc_message_t msg,
            ktc_position_t pos )
```

Set position for the message

Parameters

msg	message to set position for
pos	position

5.3.3.78 ktc_message_setRecommendationFactor()

```
void ktc_message_setRecommendationFactor (
            ktc_message_t msg,
            const char * factor,
            int value )
```

Set a recommendation factor for the message

Parameters

msg	message to set the recommendation factor for
factor	factor name
value	factor rating (0 - 100)

5.3.3.79 ktc_message_unsetFunction()

```
void ktc_message_unsetFunction ( )
```

Set current file

5.3.3.80 ktc_nodeStackGet()

```
ktc_tree_t ktc_nodeStackGet (
            int n )
```

get node from n-th position at the stack

Parameters

n	an index in the stack in the range [0; top-1] where top is a value, returned by

See also

ktc_nodeStackTop (p. 286)

Returns

node that is located at the n-th index in the stack, 0 if n is outside of range

5.3.3.81 ktc_nodeStackTop()

```
int ktc_nodeStackTop (
            void )
```

return top index of the node stack

5.3.3.82 ktc_position_copy()

```
ktc_position_t ktc_position_copy (
            ktc_position_t pos )
```

Copy previously allocated position

Parameters

position	to copy

Returns

copied position, that can be deallocated with **ktc_free()** (p. 266)

5.3.3.83 ktc_position_delete()

```
void ktc_position_delete (
            ktc_position_t pos )
```

Delete newly allocated position

5.3.3.84 ktc_position_getColumn()

```
int ktc_position_getColumn (
            ktc_position_t pos )
```

Get column from position

5.3.3.85 ktc_position_getFileName()

```
char* ktc_position_getFileName (
            ktc_position_t pos )
```

Get file name from position

5.3.3.86 ktc_position_getLine()

```
int ktc_position_getLine (
            ktc_position_t pos )
```

Get line from position

5.3.3.87 ktc_position_new()

```
ktc_position_t ktc_position_new (
            int line,
            int col,
            const char * fname )
```

Create newly allocated position with line, col and file name

Parameters

line,column	and filename to set

Returns

newly allocated position, that can be deallocated with **ktc_free()** (p. 266)

5.3.3.88 ktc_position_setLine()

```
void ktc_position_setLine (
            ktc_position_t pos,
            int line )
```

- Set line to position

5.3.3.89 ktc_proceed()

```
ktc_tree_t ktc_proceed (
            ktc_tree_t t,
            ktc_childId_t child_id )
```

From given subtree root, proceed to one of the edges designated by 'child_id'

Parameters

t	subtree root to start with
child↩ _id	child edge identifier

Returns

subtree root that is a child of t via link designated by child_id

See also

gen-ktcAPI.h (p. 41)

5.3.3.90 ktc_registerRestoreContextHook()

```
void ktc_registerRestoreContextHook (
            ktc_eventHook_t p_hook )
```

5.3.3.91 ktc_registerSaveContextHook()

```
void ktc_registerSaveContextHook (
            ktc_eventHook_t p_hook )
```

5.3.3.92 ktc_registerStartTraverseHook()

```
void ktc_registerStartTraverseHook (
            ktc_eventHook_t p_hook )
```

5.3.3.93 ktc_registerStopTraverseHook()

```
void ktc_registerStopTraverseHook (
            ktc_eventHook_t p_hook )
```

5.3.3.94 ktc_registerTreeHook()

```
void ktc_registerTreeHook (
            int tree_event,
            ktc_treeType_t tt,
            ktc_treeHook_t p_hook )
```

Register hook function for a certain kind of subtrees

Parameters

tree_event	tree event identifier
tt	tree type to hook on, may be tk_Any to hook on all tree kinds at once
p_hook	a pointer to a handler function

5.3.3.95 ktc_sema_checkBitField()

```
short ktc_sema_checkBitField (
            ktc_semanticInfo_t info )
```

Get number of bits for a bitfield

5.3.3.96 ktc_sema_findAllByName()

```
kw_array_t* ktc_sema_findAllByName (
            ktc_semanticInfo_t scope,
            const char * name )
```

Get semantic descriptions for all entities with the given name in the given scope and all its parent scopes

Parameters

scope	scope info
name	entities name

Returns

array of entity semantic descriptions

5.3.3.97 ktc_sema_findFirstByName()

```
ktc_semanticInfo_t ktc_sema_findFirstByName (
            ktc_semanticInfo_t scope,
            const char * name )
```

Get semantic information for first entity with the given name in the given scope and all its parent scopes

Parameters

scope	scope info
name	entity name

Returns

entity semantic info

5.3.3.98 ktc_sema_forAllClassDeclarations()

```
void ktc_sema_forAllClassDeclarations (
            ktc_semanticInfo_t class_info,
            void(*)( ktc_semanticInfo_t) callback )
```

Perform the given action for all declarations in the given class

Parameters

class_info	class semantic info
callback	callback performing the given action

5.3.3.99 ktc_sema_forAllScopeDeclarations()

```
void ktc_sema_forAllScopeDeclarations (
            ktc_semanticInfo_t scope_info,
            void(*)( ktc_semanticInfo_t) callback )
```

Perform the given action for all declarations in the given scope

Parameters

scope_info	scope semantic info
callback	callback performing the given action

5.3.3.100 ktc_sema_forAllSubtreeNodes()

```
void ktc_sema_forAllSubtreeNodes (
            ktc_semanticInfo_t si,
            int(*)( ktc_tree_t, void *) callback,
            void * data )
```

From subtree that corresponds to the given semantic element, apply the given callback to all subtree nodes

Parameters

si	semantic info
callback	callback to be applied The provided callback should return: 1 if no further traversal is required at all, 2 if no further children traversal is required, but Next links should be explored, and 0 otherwise. Node stack related routines should not be used from a callback

5.3.3.101 ktc_sema_functionOverloadsFunction()

```
int ktc_sema_functionOverloadsFunction (
            ktc_semanticInfo_t func1,
            ktc_semanticInfo_t func2 )
```

Check if function overloads another function in class hierarchy

Parameters

func1	first function info
func2	second function info

5.3.3.102 ktc_sema_getAllByName()

```
kw_array_t* ktc_sema_getAllByName (
            ktc_semanticInfo_t scope,
            const char * name )
```

Get semantic descriptions for all entities with the given name in the given scope

Parameters

scope	scope info
name	entities name

Returns

array of entity semantic descriptions

5.3.3.103 ktc_sema_getArrayElementType()

```
ktc_semanticInfo_t ktc_sema_getArrayElementType (
            ktc_semanticInfo_t si )
```

Get type of an array element

Parameters

si	semantic information on array type

Returns

0 if si is not an array type, semantic information on array element type otherwise

5.3.3.104 ktc_sema_getArraySize()

```
int ktc_sema_getArraySize (
            ktc_semanticInfo_t si )
```

Get size of an array

Parameters

si	semantic information on array type

Returns

-1 if si is not an array type, size of array otherwise

5.3.3.105 ktc_sema_getAST()

```
ktc_tree_t ktc_sema_getAST (
            ktc_semanticInfo_t si )
```

Get subtree that corresponds to the given semantic element

Parameters

si	semantic info

Returns

the corresponding subtree

5.3.3.106 ktc_sema_getBaseInfo()

```
ktc_semanticInfo_t ktc_sema_getBaseInfo (
            ktc_semanticInfo_t si,
            int i )
```

Get base class for current class

Parameters

i	is between 0 and **ktc_sema_getNumberOfBaseInfo()** (p. 300)-1

Returns

pointer to semantic information or 0 if no corresponding information available

5.3.3.107 ktc_sema_getBuiltinCode()

```
int ktc_sema_getBuiltinCode (
            ktc_semanticInfo_t si )
```

Get builtin type code by code item semantic information

Parameters

si	semantic description of the code item

Returns

operation code (see **Numerical codes of C/C++ built-in types** (p. 34)) or KTC_BUILTINTYPE_NONE if the item is not of builtin type

5.3.3.108 ktc_sema_getClassTag()

```
int ktc_sema_getClassTag (
            ktc_semanticInfo_t si )
```

Get class tag code for a type described by provided semantic information

5.3.3.109 ktc_sema_getCVQualifiers()

```
int ktc_sema_getCVQualifiers (
            ktc_semanticInfo_t si )
```

Get const and volatile qualifiers of an entity described by 'si'

5.3.3.110 ktc_sema_getDefinedType()

```
ktc_semanticInfo_t ktc_sema_getDefinedType (
            ktc_semanticInfo_t si )
```

Get type defined by typedef alias

Parameters

si	semantic information on typedef alias

Returns

0 if si is not a typedef alias, semantic information on typedef'd type otherwise

5.3.3.111 ktc_sema_getEnumerators()

```
kw_array_t* ktc_sema_getEnumerators (
            ktc_semanticInfo_t si )
```

Get actual enumerators for a given enum

Parameters

si	semantic description of enum

Returns

array of enumerators' semantic descriptions

5.3.3.112 ktc_sema_getEnumUnderlyingTypeOrInt()

```
ktc_semanticInfo_t ktc_sema_getEnumUnderlyingTypeOrInt (
            ktc_semanticInfo_t info )
```

Returns

the semantic information describing the type underlying the given enum. If the underlying type is not known, returns type information for int.

5.3.3.113 ktc_sema_getFieldNumber()

```
int ktc_sema_getFieldNumber (
            ktc_semanticInfo_t si )
```

Get number of struct fields

Parameters

struct	semantic info

Returns

0 if the semantic info does not correspond to the struct, number of fields otherwise

5.3.3.114 ktc_sema_getFieldType()

```
ktc_semanticInfo_t ktc_sema_getFieldType (
            ktc_semanticInfo_t si,
            int fieldpos )
```

Get type of struct or class field

Parameters

struct	or class semantic info, number of field

Returns

0 if the semantic info does not correspond to the struct, the semantic type of the field otherwise

5.3.3.115 ktc_sema_getFirstByName()

```
ktc_semanticInfo_t ktc_sema_getFirstByName (
            ktc_semanticInfo_t scope,
           const char * name )
```

Get semantic information for first entity with the given name in the given scope

Parameters

scope	scope info
name	entity name

Returns

entity semantic info

5.3.3.116 ktc_sema_getFormalArgument()

```
ktc_semanticInfo_t ktc_sema_getFormalArgument (
            ktc_semanticInfo_t si,
           int n )
```

Get semantic information of the n-th function type formal argument

Parameters

si	semantic description of the function type
n	formal parameter number. Must be in a range [0;N), where N - number returned by function 'ktc_sema_getNumberOfArguments'. Argument number 0 is non-zero only for class methods and desribes implicit this argument

Returns

semantic description of n-th formal argument, or 0 if 'si' is not a function type description.

5.3.3.117 ktc_sema_getFunctionFromTemplate()

```
ktc_semanticInfo_t ktc_sema_getFunctionFromTemplate (
            ktc_semanticInfo_t info )
```

Get function from semantic information about function template

Returns

semantic information or 0 if si is not a function template description

5.3.3.118 ktc_sema_getFunctionPointerType()

```
ktc_semanticInfo_t ktc_sema_getFunctionPointerType (
            ktc_semanticInfo_t info )
```

Returns

the semantic information decribing a function pointer type, 0 if not found.

5.3.3.119 ktc_sema_getFunctionTemplateInstantiations()

```
kw_array_t* ktc_sema_getFunctionTemplateInstantiations (
            ktc_semanticInfo_t info )
```

Get all instantiations for a function template

5.3.3.120 ktc_sema_getFunctionTemplateSpecializations()

```
kw_array_t* ktc_sema_getFunctionTemplateSpecializations (
            ktc_semanticInfo_t info )
```

Get all specializations for a function template

5.3.3.121 ktc_sema_getFunctionType()

```
ktc_semanticInfo_t ktc_sema_getFunctionType (
            ktc_semanticInfo_t si )
```

Get function type type from a function semantic description

Returns

function type if si is a semantic description of a function or 0 otherwise

5.3.3.122 ktc_sema_getGlobalScope()

```
ktc_semanticInfo_t ktc_sema_getGlobalScope ( )
```

Get semantic information for the global scope

Returns

global scope semantic info

5.3.3.123 ktc_sema_getIdentifier()

```
const char* ktc_sema_getIdentifier (
            ktc_semanticInfo_t si )
```

Get semantic entity name

5.3.3.124 ktc_sema_getIdentifierNo()

```
int ktc_sema_getIdentifierNo (
            ktc_semanticInfo_t si )
```

5.3.3.125 ktc_sema_getInstantiatedClass()

```
ktc_semanticInfo_t ktc_sema_getInstantiatedClass (
            ktc_semanticInfo_t si )
```

Get class semantic info for an instantiation

5.3.3.126 ktc_sema_getInstantiationOrigin()

```
ktc_semanticInfo_t ktc_sema_getInstantiationOrigin (
            ktc_semanticInfo_t info )
```

Get a specialization the given instantiation was created from

5.3.3.127 ktc_sema_getInstantiationParameters()

```
kw_array_t* ktc_sema_getInstantiationParameters (
            ktc_semanticInfo_t si )
```

Get actual template parameters for a given instantiation

Parameters

si	semantic description of an instantiation

Returns

array of parameters' semantic descriptions

5.3.3.128 ktc_sema_getIntegerValue()

```
ktc_long_long_t ktc_sema_getIntegerValue (
            ktc_semanticInfo_t val,
            int * error_flag )
```

Get integer value of constant of one of integer types or bool type (in C++)

Parameters

val	semantic info
error_flag	set to non-zero value if it is impossible to convert constant into integer value

Returns

integer value of a constant, 0 is returned for 'false', 1 is returned for 'true'

5.3.3.129 ktc_sema_getIntValueType()

```
int ktc_sema_getIntValueType (
            ktc_semanticInfo_t si )
```

Get the original type of an int value

Parameters

value	semantic info

Returns

the KTC_BUILTIN type code

5.3.3.130 ktc_sema_getNumber()

```
int ktc_sema_getNumber (
            ktc_semanticInfo_t si )
```

5.3.3.131 ktc_sema_getNumberOfArguments()

```
int ktc_sema_getNumberOfArguments (
            ktc_semanticInfo_t si )
```

Get number of arguments from function type

Returns

number of formal arguments for a given function type. 0 means that function type describes method with 'this' as only (and implicit) parameter. If 'si; is not a function type, -1 is returned.

5.3.3.132 ktc_sema_getNumberOfBaseInfo()

```
int ktc_sema_getNumberOfBaseInfo (
            ktc_semanticInfo_t si )
```

Get number of base classes for current class

Returns

number of base classes

5.3.3.133 ktc_sema_getObjectName()

```
const char* ktc_sema_getObjectName (
            ktc_semanticInfo_t si )
```

Get object name string of an object value

Parameters

object	semantic info

Returns

the name string or NULL on error. The string pointer should not be freed.

5.3.3.134 ktc_sema_getOverridenMethod()

```
ktc_semanticInfo_t ktc_sema_getOverridenMethod (
            ktc_tree_t t )
```

Get semantic description of a base class method overriden with method that is associated with current node. Applicable to nodes of the following kinds: MemberDecl, MemberFunc, FuncDef

Returns

pointer to semantic information or 0 if no corresponding information available

5.3.3.135 ktc_sema_getPointedType()

```
ktc_semanticInfo_t ktc_sema_getPointedType (
            ktc_semanticInfo_t si )
```

Get type pointed by a pointer type

Parameters

si	semantic information on pointer type

Returns

0 if si is not a pointer type, semantic information on pointed type otherwise

5.3.3.136 ktc_sema_getPosition()

```
ktc_position_t ktc_sema_getPosition (
            ktc_semanticInfo_t si )
```

Get position of the declaration that corresponds to the given semantic element

Parameters

si	semantic info

Returns

the corresponding position

5.3.3.137 ktc_sema_getPrimaryTemplate()

```
ktc_semanticInfo_t ktc_sema_getPrimaryTemplate (
            ktc_semanticInfo_t si )
```

Get primary template for the given instantiation or specialization

Parameters

si	semantic description of an instantiation or a specialization

Returns

semantic description of the primary template

5.3.3.138 ktc_sema_getQualifiedName()

```
char* ktc_sema_getQualifiedName (
            ktc_semanticInfo_t si )
```

Get fully qualified name

Returns

newly allocated string, that can be deallocated with **ktc_free()** (p. 266)

5.3.3.139 ktc_sema_getReferencedType()

```
ktc_semanticInfo_t ktc_sema_getReferencedType (
            ktc_semanticInfo_t si )
```

Get type referenced by a reference type (C++ specific)

Parameters

si	semantic information on reference type

Returns

0 if si is not a reference type, semantic information on referenced type otherwise

5.3.3.140 ktc_sema_getReturnType()

```
ktc_semanticInfo_t ktc_sema_getReturnType (
            ktc_semanticInfo_t si )
```

Get function return type from function type semantic information

Returns

function return type if si is semantic information on function type or 0 if si is not a function type description

5.3.3.141 ktc_sema_getScope()

```
ktc_semanticInfo_t ktc_sema_getScope (
            ktc_semanticInfo_t si )
```

Get semantic description of class for current node.

Returns

pointer to semantic information or 0 if no corresponding information available

5.3.3.142 ktc_sema_getTypedefedName()

```
const char* ktc_sema_getTypedefedName (
            ktc_semanticInfo_t si )
```

Get name defined via typedef for anonymous structs and enums

5.3.3.143 ktc_sema_getTypeName()

```
char* ktc_sema_getTypeName (
            ktc_semanticInfo_t si )
```

Get a newly allocated string containing name of language type

Returns

newly allocated string, that can be deallocated with **ktc_free()** (p. 266)

5.3.3.144 ktc_sema_getVariableInitializer()

```
ktc_tree_t ktc_sema_getVariableInitializer (
            ktc_semanticInfo_t si )
```

Get variable initializer from variable semantic information

Returns

tree node on variable initializer or 0 if si is not a variable description / does not have initializer

5.3.3.145 ktc_sema_getVariableType()

```
ktc_semanticInfo_t ktc_sema_getVariableType (
            ktc_semanticInfo_t si )
```

Get variable type from variable semantic information

Returns

semantic information on variable type or 0 if si is not a variable description

5.3.3.146 ktc_sema_getVariableValue()

```
ktc_semanticInfo_t ktc_sema_getVariableValue (
            ktc_semanticInfo_t si )
```

Get variable value from variable semantic information

Returns

semantic information on variable value or 0 if si is not a variable description

5.3.3.147 ktc_sema_hasMethods()

```
int ktc_sema_hasMethods (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a class with declared methods

5.3.3.148 ktc_sema_haveSameFunctionType()

```
int ktc_sema_haveSameFunctionType (
            ktc_semanticInfo_t si1,
            ktc_semanticInfo_t si2 )
```

Check if function types are the close (regardless of the scope) Applicable to function types

Returns

1 if functions are the close, 0 otherwise

5.3.3.149 ktc_sema_haveSameSignature()

```
int ktc_sema_haveSameSignature (
            ktc_semanticInfo_t si1,
            ktc_semanticInfo_t si2 )
```

Check if functions are the close (regardless of the scope) Applicable to descriptors of functions

Returns

1 if functions are the close, 0 otherwise

5.3.3.150 ktc_sema_isAnonymous()

```
int ktc_sema_isAnonymous (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of an anonymous struct

5.3.3.151 ktc_sema_isArray()

```
int ktc_sema_isArray (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of array

5.3.3.152 ktc_sema_isBasePrivate()

```
int ktc_sema_isBasePrivate (
            ktc_semanticInfo_t si,
           int i )
```

Check if particular base of the given class is private

Parameters

si	derived class info
i	is between 0 and **ktc_sema_getNumberOfBaseInfo()** (p. 300)-1

Returns

1 if base is private, 0 otherwise

5.3.3.153 ktc_sema_isBaseProtected()

```
int ktc_sema_isBaseProtected (
            ktc_semanticInfo_t si,
            int i )
```

Check if particular base of the given class is protected

Parameters

si	derived class info
i	is between 0 and **ktc_sema_getNumberOfBaseInfo()** (p. 300)-1

Returns

1 if base is protected, 0 otherwise

5.3.3.154 ktc_sema_isBasePublic()

```
int ktc_sema_isBasePublic (
            ktc_semanticInfo_t si,
            int i )
```

Check if particular base of the given class is public

Parameters

si	derived class info
i	is between 0 and **ktc_sema_getNumberOfBaseInfo()** (p. 300)-1

Returns

1 if base is public, 0 otherwise

5.3.3.155 ktc_sema_isBaseVirtual()

```
int ktc_sema_isBaseVirtual (
            ktc_semanticInfo_t si,
            int i )
```

Check if particular base of the given class is virtual

Parameters

si	derived class info
i	is between 0 and **ktc_sema_getNumberOfBaseInfo()** (p. 300)-1

Returns

1 if base is virtual, 0 otherwise

5.3.3.156 ktc_sema_isBitfield()

```
int ktc_sema_isBitfield (
            ktc_semanticInfo_t si )
```

Check whether symbol described by semantic information is a bit field in a structure

Parameters

si	semantic information handle

Returns

0 if symbol is not a bit field, number of bits if symbol is a bit field

5.3.3.157 ktc_sema_isBuiltin()

```
int ktc_sema_isBuiltin (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of built-in type

5.3.3.158 ktc_sema_isClass()

```
int ktc_sema_isClass (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of class

5.3.3.159 ktc_sema_isConstMethod()

```
int ktc_sema_isConstMethod (
            ktc_semanticInfo_t si )
```

Check if particular method is constant

Parameters

si	method info

Returns

1 if method is constant, 0 otherwise

5.3.3.160 ktc_sema_isConstructor()

```
int ktc_sema_isConstructor (
            ktc_semanticInfo_t si )
```

Check whether given semantic info describes a class constructor

5.3.3.161 ktc_sema_isDestructor()

```
int ktc_sema_isDestructor (
            ktc_semanticInfo_t si )
```

Check whether given semantic info describes a class destructor

5.3.3.162 ktc_sema_isEnum()

```
int ktc_sema_isEnum (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of enum

5.3.3.163 ktc_sema_isEnumConstant()

```
int ktc_sema_isEnumConstant (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of an enumeration constant

5.3.3.164 ktc_sema_isFriend()

```
int ktc_sema_isFriend (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a friend element

5.3.3.165 ktc_sema_isFuncDef()

```
int ktc_sema_isFuncDef (
            ktc_semanticInfo_t info )
```

Check that given node corresponds to a function definition

Returns

1 if true; 0 if false

5.3.3.166 ktc_sema_isFunction()

```
int ktc_sema_isFunction (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of function

5.3.3.167 ktc_sema_isFunctionPointer()

```
int ktc_sema_isFunctionPointer (
            ktc_semanticInfo_t info )
```

Check if the given semantic information decribes a function pointer

5.3.3.168 ktc_sema_isFunctionTemplate()

```
int ktc_sema_isFunctionTemplate (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of function template

5.3.3.169 ktc_sema_isFunctionTemplateSet()

```
int ktc_sema_isFunctionTemplateSet (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of function template set

5.3.3.170 ktc_sema_isFunctionType()

```
int ktc_sema_isFunctionType (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of function type

5.3.3.171 ktc_sema_isGlobal()

```
int ktc_sema_isGlobal (
            ktc_semanticInfo_t si )
```

Check whether given semantic info describes an entity declared in the global scope

5.3.3.172 ktc_sema_isImmutable()

```
int ktc_sema_isImmutable (
            ktc_semanticInfo_t si )
```

Check if an entity described by 'si' is immutable (has const qualifier or is an array)

Parameters

si	semantic description of the code item

Returns

1 for immutable items, 0 otherwise

5.3.3.173 ktc_sema_isIncluded()

```
int ktc_sema_isIncluded (
            ktc_semanticInfo_t si )
```

Check that given sema info corresponds to an included entity

Returns

1 if is; 0 otherwise

5.3.3.174 ktc_sema_isInline()

```
int ktc_sema_isInline (
            ktc_semanticInfo_t info )
```

Check if the given function is inline

5.3.3.175 ktc_sema_isInstantiatedFunction()

```
int ktc_sema_isInstantiatedFunction (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of an instantiated function

5.3.3.176 ktc_sema_isInstantiation()

```
int ktc_sema_isInstantiation (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of instantiation

5.3.3.177 ktc_sema_isIntegerValue()

```
int ktc_sema_isIntegerValue (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of integervalue

5.3.3.178 ktc_sema_isLocal()

```
int ktc_sema_isLocal (
            ktc_semanticInfo_t si )
```

Check whether given semantic info describes an entity declared in a local scope

5.3.3.179 ktc_sema_isNamespace()

```
int ktc_sema_isNamespace (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of namespace

5.3.3.180 ktc_sema_isNamespaceAlias()

```
int ktc_sema_isNamespaceAlias (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of namespace alias

5.3.3.181 ktc_sema_isNone()

```
int ktc_sema_isNone (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of none

5.3.3.182 ktc_sema_isObjectValue()

```
int ktc_sema_isObjectValue (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of objectvalue

5.3.3.183 ktc_sema_isOperatorFunction()

```
int ktc_sema_isOperatorFunction (
            ktc_semanticInfo_t si )
```

Check whether given semantic info describes an operator function

5.3.3.184 ktc_sema_isPOD()

```
int ktc_sema_isPOD (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a POD type

5.3.3.185 ktc_sema_isPointer()

```
int ktc_sema_isPointer (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of pointer

5.3.3.186 ktc_sema_isPrivate()

```
int ktc_sema_isPrivate (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a private class member

5.3.3.187 ktc_sema_isProtected()

```
int ktc_sema_isProtected (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a protected class member

5.3.3.188 ktc_sema_isPublic()

```
int ktc_sema_isPublic (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a public class member

5.3.3.189 ktc_sema_isPureVirtual()

```
int ktc_sema_isPureVirtual (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a pure virtual class method

5.3.3.190 ktc_sema_isReference()

```
int ktc_sema_isReference (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of reference

5.3.3.191 ktc_sema_isSameClass()

```
int ktc_sema_isSameClass (
            ktc_semanticInfo_t si1,
            ktc_semanticInfo_t si2 )
```

Check if classes are the same Applicable to descriptors of classes

Returns

1 if classes are the same, 0 if classes are different

5.3.3.192 ktc_sema_isSameEnum()

```
int ktc_sema_isSameEnum (
            ktc_semanticInfo_t si1,
            ktc_semanticInfo_t si2 )
```

Check if enums are the same Applicable to descriptors of enums

Returns

1 if enums are the same, 0 if enums are different

5.3.3.193 ktc_sema_isSameFunctions()

```
int ktc_sema_isSameFunctions (
            ktc_semanticInfo_t si1,
            ktc_semanticInfo_t si2 )
```

Check if functions are the same Applicable to descriptors of functions

Returns

1 if functions are the same, 0 if functions are different

5.3.3.194 ktc_sema_isSameScope()

```
int ktc_sema_isSameScope (
            ktc_semanticInfo_t si1,
            ktc_semanticInfo_t si2 )
```

Check if scopes are the same Applicable to descriptors of scopes

Returns

1 if scopes are the same, 0 if scopes are different

5.3.3.195 ktc_sema_isSameVariables()

```
int ktc_sema_isSameVariables (
            ktc_semanticInfo_t si1,
            ktc_semanticInfo_t si2 )
```

Check if variables are the same Applicable to descriptors of variables

Returns

1 if variables are the same, 0 if variables are different

5.3.3.196 ktc_sema_isScope()

```
int ktc_sema_isScope (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of scope

5.3.3.197 ktc_sema_isSpecialization()

```
int ktc_sema_isSpecialization (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of specialization

5.3.3.198 ktc_sema_isStatic()

```
int ktc_sema_isStatic (
            ktc_semanticInfo_t info )
```

Check that given entity corresponds to a static declaration

Returns

1 if true, 0 if false

5.3.3.199 ktc_sema_isTemplate()

```
int ktc_sema_isTemplate (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of template

5.3.3.200 ktc_sema_isType()

```
int ktc_sema_isType (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of type

5.3.3.201 ktc_sema_isTypeParameter()

```
int ktc_sema_isTypeParameter (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of a template type parameter

5.3.3.202 ktc_sema_isUnion()

```
int ktc_sema_isUnion (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of union

5.3.3.203 ktc_sema_isUsing()

```
int ktc_sema_isUsing (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of using

5.3.3.204 ktc_sema_isUsingDeclaration()

```
int ktc_sema_isUsingDeclaration (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of using declaration

5.3.3.205 ktc_sema_isUsingDirective()

```
int ktc_sema_isUsingDirective (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of using directive

5.3.3.206 ktc_sema_isVariable()

```
int ktc_sema_isVariable (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of variable

5.3.3.207 ktc_sema_isVirtual()

```
int ktc_sema_isVirtual (
            ktc_semanticInfo_t si )
```

Check whether semantic information is a description of virtual method

5.3.3.208 ktc_sema_RelatedClasses()

```
int ktc_sema_RelatedClasses (
            ktc_semanticInfo_t class1,
            ktc_semanticInfo_t class2 )
```

Check if two class are related through inheritance relationship

Parameters

class1	first class info
class2	second class info

5.3.3.209 ktc_sema_skipTypedefs()

```
ktc_semanticInfo_t ktc_sema_skipTypedefs (
            ktc_semanticInfo_t si,
           int * cvq )
```

Get type defined by all typedef alias

Parameters

si	semantic information on typedef alias
cvq	if not NULL saves const and volatile qualifiers

Returns

si if si is not a typedef alias, semantic information on all typedef'd types otherwise

5.3.3.210 ktc_sema_stripCVQ()

```
ktc_semanticInfo_t ktc_sema_stripCVQ (
            ktc_semanticInfo_t si )
```

Get type identical to the given one but without 'const' and 'volatile' qualifiers

5.3.3.211 ktc_skipBrackets()

```
ktc_tree_t ktc_skipBrackets (
            ktc_tree_t t )
```

Drop redundant parentheses around expression in expression tree

Returns

a node that is a root of subtree of first non-parenthesized expression, for example, ((((a+(b)))) will be traversed down to a+(b)

5.3.3.212 ktc_string_delete()

```
void ktc_string_delete (
            ktc_string_t ks )
```

Delete ktc string

Parameters

ks	ktc string

5.3.3.213 ktc_string_get_cstring()

```
const char* ktc_string_get_cstring (
            ktc_string_t ks )
```

Get character string from ktc string

Parameters

ks	ktc string

Returns

character string

5.3.3.214 ktc_string_new()

```
ktc_string_t ktc_string_new (
            const char * s )
```

Create new ktc string from character string

Parameters

s	character string

Returns

newly created ktc string

5.3.3.215 ktc_treeType_getName()

```
const char* ktc_treeType_getName (
             ktc_tree_t ttype )
```

Get name of tree type for particular tree vertex (without 'tid_' prefix). Mainly for debug purposes.

5.3.3.216 ktc_unregisterTreeHook()

```
void ktc_unregisterTreeHook (
            int tree_event,
             ktc_treeType_t tt,
             ktc_treeHook_t p_hook )
```

Deregister hook from a hook table

Parameters

tree_event	tree event identifier
tt	tree type
p_hook	a hook address to deregister

5.3.4 Variable Documentation

5.3.4.1 ktc_constructorName

```
const char* ktc_constructorName
```

5.3.4.2 ktc_destructorName

```
const char* ktc_destructorName
```

5.3.4.3 KTC_DIALECT_ARM

```
const int KTC_DIALECT_ARM
```

Current translation unit is compiled in armcc compiler emulation mode

5.3.4.4 KTC_DIALECT_GHS

```
const int KTC_DIALECT_GHS
```

Current translation unit is compiled in Green Hills compiler emulation mode

5.3.4.5 KTC_DIALECT_GNU

```
const int KTC_DIALECT_GNU
```

Current translation unit is compiled in gnu emulation mode

5.3.4.6 KTC_DIALECT_MS

```
const int KTC_DIALECT_MS
```

Current translation unit is compiled in MS cl emulation mode

5.3.4.7 KTC_DIALECT_STD

```
const int KTC_DIALECT_STD
```

Current translation unit is compiled in standard mode

5.3.4.8 KTC_DIALECT_SUN

```
const int KTC_DIALECT_SUN
```

Current translation unit is compiled in Sun compiler emulation mode

5.3.4.9 KTC_DIALECT_TI

```
const int KTC_DIALECT_TI
```

Current translation unit is compiled in Texas Instruments compiler emulation mode

5.3.4.10 KTC_LANGUAGE_C

```
const int KTC_LANGUAGE_C
```

Current translation unit is compiled by C frontend

5.3.4.11 KTC_LANGUAGE_CXX

```
const int KTC_LANGUAGE_CXX
```

Current translation unit is compiled by C++ frontend

5.3.4.12 KTC_TREE_EVENT_ON_ENTER

```
int KTC_TREE_EVENT_ON_ENTER
```

'On Enter' traversal event identifier. Defines an event when some node is entered during depth-first traversal

5.3.4.13 KTC_TREE_EVENT_ON_LEAVE

```
int KTC_TREE_EVENT_ON_LEAVE
```

'On Leave' traversal event identifier. Defines an event when all links of a node (including Next) were traversed and traversal is about to enter a sibling node

5.3.4.14 KTC_TREE_EVENT_ON_NEXT

```
int KTC_TREE_EVENT_ON_NEXT
```

'On Next' traversal event identifier. Defines an event when some node is left via 'Next' link to the next node in the list of nodes. Next link is always processed after all children's subtrees were fully traversed

5.3.4.15 KTC_TYPESIGNEDNESS_DEFAULT

```
int KTC_TYPESIGNEDNESS_DEFAULT
```

Default value for 'int', 'char', 'short', 'long' 'long long'

5.3.4.16 KTC_TYPESIGNEDNESS_NONE

```
int KTC_TYPESIGNEDNESS_NONE
```

Type is neither signed nor unsigned. All built-in non-integer types and all non built-in types do not have signedness

5.3.4.17 KTC_TYPESIGNEDNESS_SIGNED

```
int KTC_TYPESIGNEDNESS_SIGNED
```

Value for 'signed int|char|short|long|long long'

5.3.4.18 KTC_TYPESIGNEDNESS_UNSIGNED

```
int KTC_TYPESIGNEDNESS_UNSIGNED
```

Value for 'unsigned int|char|short|long|long long'

5.4 kwapi.h File Reference

```
#include <stddef.h>
```

Macros

- #define **KWAPI_DECLARE**(type) type
- #define **KWAPI_DECLARE_CPP**(type) type
- #define **KWAPI_DECLARE_NONSTD**(type) type
- #define **KWAPI_DECLARE_DATA**

Typedefs

- typedef size_t **kw_size_t**
- typedef struct ParameterNode ∗ **kwapi_cfgparam_t**
- typedef struct **kw_array_t** **kw_array_t**

Enumerations

- enum **kwapi_apitypes_t** {
  KWAPI_NONE = 0x0, **KWAPI_TREE** = 0x1, **KWAPI_PATH** = 0x2, **KWAPI_PATTERN** = 0x4,
  KWAPI_PREP = 0x8, **KWAPI_LINKER** = 0x10 }
- enum **kwapi_langtypes_t** { **KWAPI_NOLANG**, **KWAPI_JAVA**, **KWAPI_CXX**, **KWAPI_CSHARP** }

Functions

- **kwapi_cfgparam_t kwapi_cfgparam_getRootParameterList** (const char ∗error)
- **kwapi_cfgparam_t kwapi_cfgparam_getListNodeByName** (**kwapi_cfgparam_t**, const char ∗name)
- **kwapi_cfgparam_t kwapi_cfgparam_getListNodeByRegexMatchingName** (**kwapi_cfgparam_t**, const char ∗name)
- const char ∗ **kwapi_cfgparam_getName** (**kwapi_cfgparam_t**)
- const char ∗ **kwapi_cfgparam_getType** (**kwapi_cfgparam_t**)
- **kw_size_t kwapi_cfgparam_getListLength** (**kwapi_cfgparam_t**)
- **kwapi_cfgparam_t kwapi_cfgparam_getListNodeByIndex** (**kwapi_cfgparam_t**, **kw_size_t** idx)
- int **kwapi_cfgparam_isParameter** (**kwapi_cfgparam_t**)
- const char ∗ **kwapi_cfgparam_getParameterValue** (**kwapi_cfgparam_t**)
- const char ∗ **kwapi_cfgparam_getParameterValueFromList** (**kwapi_cfgparam_t** parent, const char ∗paramName)
- const char ∗ **kwapi_cfgparam_getConfigurationParameter** (const char ∗errorId, const char ∗paramName)
- const char ∗const ∗ **kwapi_cfgparam_getCheckerErrors** (const char ∗checker_id)
- const char ∗ **ktc_error_getConfigurationParameter** (const char ∗errorId, const char ∗paramName)
- int **kwapi_cfgparam_errorIsEnabled** (const char ∗error_id)
- **kw_size_t kw_array_size** (**kw_array_t** ∗array)
- void ∗ **kw_array_get** (**kw_array_t** ∗array, **kw_size_t** index)
- void **kw_array_delete** (**kw_array_t** ∗array)

5.4.1 Macro Definition Documentation

5.4.1.1 KWAPI_DECLARE

```
#define KWAPI_DECLARE(
              type ) type
```

5.4.1.2 KWAPI_DECLARE_CPP

```
#define KWAPI_DECLARE_CPP(
              type ) type
```

5.4.1.3 KWAPI_DECLARE_DATA

```
#define KWAPI_DECLARE_DATA
```

5.4.1.4 KWAPI_DECLARE_NONSTD

```
#define KWAPI_DECLARE_NONSTD(
              type ) type
```

5.4.2 Typedef Documentation

5.4.2.1 kw_array_t

```
typedef struct kw_array_t kw_array_t
```

5.4.2.2 kw_size_t

```
typedef size_t kw_size_t
```

5.4.2.3 kwapi_cfgparam_t

```
typedef struct ParameterNode* kwapi_cfgparam_t
```

5.4.3 Enumeration Type Documentation

5.4.3.1 kwapi_apitypes_t

```
enum kwapi_apitypes_t
```

Constants for Klocwork APIs

Enumerator

KWAPI_NONE	
KWAPI_TREE	
KWAPI_PATH	
KWAPI_PATTERN	
KWAPI_PREP	
KWAPI_LINKER	

5.4.3.2 kwapi_langtypes_t

```
enum kwapi_langtypes_t
```

Constants for Klocwork supported languages

Enumerator

KWAPI_NOLANG	
KWAPI_JAVA	
KWAPI_CXX	
KWAPI_CSHARP	

5.4.4 Function Documentation

5.4.4.1 ktc_error_getConfigurationParameter()

```
const char* ktc_error_getConfigurationParameter (
            const char * errorId,
            const char * paramName )
```

5.4.4.2 kw_array_delete()

```
void kw_array_delete (
            kw_array_t * array )
```

5.4.4.3 kw_array_get()

```
void* kw_array_get (
            kw_array_t * array,
            kw_size_t index )
```

5.4.4.4 kw_array_size()

```
kw_size_t kw_array_size (
            kw_array_t * array )
```

5.4.4.5 kwapi_cfgparam_errorIsEnabled()

```
int kwapi_cfgparam_errorIsEnabled (
            const char * error_id )
```

Check that particular error was enabled in the configuration file

Parameters

error↩ _id	identifier of an error (the same id as in <error> tag in configuration file)

Returns

0 if error was disabled or was not present in the configuration file, 1 otherwise

5.4.4.6 kwapi_cfgparam_getCheckerErrors()

```
const char* const* kwapi_cfgparam_getCheckerErrors (
            const char * checker_id )
```

Obtain a pointer to the internal array of error ids that are produced by a given checker

Parameters

checker↩ _id	Identifier of a checker

Returns

a pointer to a zero terminated array of strings containing error identifiers or 0 if checker was not configured

5.4.4.7 kwapi_cfgparam_getConfigurationParameter()

```
const char* kwapi_cfgparam_getConfigurationParameter (
            const char * errorId,
            const char * paramName )
```

Quick access to parameters for errors that do not require parameterlist'.

Remarks

Essentialy gets a parameter by name from a root list `kwapi_cfgparam_t el = kwapi_cfgparam↩ _getRootParameterList(errorId); if (el) { return kwapi_cfgparam_get↩ ParameterValueFromList(el, paramName); } else { return 0; }`

5.4.4.8 kwapi_cfgparam_getListLength()

```
kw_size_t kwapi_cfgparam_getListLength (
            kwapi_cfgparam_t  )
```

Returns 0 for parameter's, list length for parameterlist's

5.4.4.9 kwapi_cfgparam_getListNodeByIndex()

```
kwapi_cfgparam_t kwapi_cfgparam_getListNodeByIndex (
            kwapi_cfgparam_t ,
            kw_size_t idx )
```

Returns 0 for parameters, parameter node at index idx for lists, or 0 if index is outside of bounds

5.4.4.10 kwapi_cfgparam_getListNodeByName()

```
kwapi_cfgparam_t kwapi_cfgparam_getListNodeByName (
            kwapi_cfgparam_t ,
           const char * name )
```

Extract parameter node from a list by name,

Returns

0 if there is no parameter node with such name, or parameter node is not a parameterlist

5.4.4.11 kwapi_cfgparam_getListNodeByRegexMatchingName()

```
kwapi_cfgparam_t kwapi_cfgparam_getListNodeByRegexMatchingName (
            kwapi_cfgparam_t ,
           const char * name )
```

Extract parameter node from a list by name, where the keys associated with nodes are treated as PERL-compatible regular expressions (`https://www.pcre.org`) when verifying if name matches a list entry. Specifically, when kwapi_cfgparam_t denotes a ParameterList (i.e., a list of key-value pairs whose key is of type string and value is of type ParameterNode), this function will return the ParameterNode of the element e in kwapi_cfgparam_t where e's key is a PERL-compatible regular expression, and this regular expression matches name in its entirety. If there are multiple elements in kwapi_cfgparam_t whose key matches name, then the ParameterNode of the first element encountered is returned. Otherwise:

Returns

0 if there is no parameter node in list kwapi_cfgparam_t whose key is a PERL-compatible regular expression that matches name, or parameter node is not a parameterlist

5.4.4.12 kwapi_cfgparam_getName()

```
const char* kwapi_cfgparam_getName (
            kwapi_cfgparam_t  )
```

Get name of parameter node, or 0 if parameter has no name

5.4.4.13 kwapi_cfgparam_getParameterValue()

```
const char* kwapi_cfgparam_getParameterValue (
            kwapi_cfgparam_t  )
```

Return parameter string value for parameters, 0 for parameterlists

5.4.4.14 kwapi_cfgparam_getParameterValueFromList()

```
const char* kwapi_cfgparam_getParameterValueFromList (
            kwapi_cfgparam_t parent,
           const char * paramName )
```

Get parameter value from a parameter in the list by parameter name

Remarks

essentially a code: `kwapi_cfgparam_t el = kwapi_cfgparam_getListNodeByName(parent); if (el) { return kwapi_cfgparam_getParameterValue(paramName); } else { return 0; }`

5.4.4.15 kwapi_cfgparam_getRootParameterList()

```
kwapi_cfgparam_t kwapi_cfgparam_getRootParameterList (
           const char * error )
```

Get handler of root parameter list: it contains all parameter's and parameterlist's at the top level of <error> tag

5.4.4.16 kwapi_cfgparam_getType()

```
const char* kwapi_cfgparam_getType (
            kwapi_cfgparam_t  )
```

Get type of parameter node , or 0 if there is no type specified

5.4.4.17 kwapi_cfgparam_isParameter()

```
int kwapi_cfgparam_isParameter (
            kwapi_cfgparam_t  )
```

Returns 1 if node is parameter, 0 if it is parameterlist

Index

Access to semantic information, 11
Accessing compiler configuration, 17
Accessing node stack, 9
Attribute values for signedness of types, 13

Basic Abstract Syntax Tree traversal and checking routines, 8

Child link identifiers, 21
cid_Adjacent
    gen-ktcAPI.h, 215
cid_Args
    gen-ktcAPI.h, 216
cid_Attributes
    gen-ktcAPI.h, 216
cid_AttributeSpec
    gen-ktcAPI.h, 216
cid_AttributeSpecs
    gen-ktcAPI.h, 216
cid_Base
    gen-ktcAPI.h, 216
cid_BaseSpecs
    gen-ktcAPI.h, 216
cid_Bits
    gen-ktcAPI.h, 216
cid_Cond
    gen-ktcAPI.h, 216
cid_ConversionType
    gen-ktcAPI.h, 217
cid_CtorInit
    gen-ktcAPI.h, 217
cid_CVQualifiers
    gen-ktcAPI.h, 217
cid_Decl
    gen-ktcAPI.h, 217
cid_Declarator
    gen-ktcAPI.h, 217
cid_Declarators
    gen-ktcAPI.h, 217
cid_Decls
    gen-ktcAPI.h, 217
cid_DeclSpecs
    gen-ktcAPI.h, 217
cid_Default
    gen-ktcAPI.h, 218
cid_Designators
    gen-ktcAPI.h, 218
cid_Else
    gen-ktcAPI.h, 218
cid_Enumerators
    gen-ktcAPI.h, 218
cid_Exception
    gen-ktcAPI.h, 218
cid_Expr
    gen-ktcAPI.h, 218
cid_Func
    gen-ktcAPI.h, 218
cid_FuncBody
    gen-ktcAPI.h, 218
cid_Handler
    gen-ktcAPI.h, 219
cid_Handlers
    gen-ktcAPI.h, 219
cid_Index
    gen-ktcAPI.h, 219
cid_Init
    gen-ktcAPI.h, 219
cid_Initializer
    gen-ktcAPI.h, 219
cid_Inits
    gen-ktcAPI.h, 219
cid_Introducer
    gen-ktcAPI.h, 219
cid_KRParams
    gen-ktcAPI.h, 219
cid_Label
    gen-ktcAPI.h, 220
cid_LambdaCapture
    gen-ktcAPI.h, 220
cid_Left
    gen-ktcAPI.h, 220
cid_Literal
    gen-ktcAPI.h, 220
cid_Lower
    gen-ktcAPI.h, 220
cid_MemberDecl
    gen-ktcAPI.h, 220
cid_MemberDecls
    gen-ktcAPI.h, 220
cid_MemberInitializers
    gen-ktcAPI.h, 220
cid_Name
    gen-ktcAPI.h, 221
cid_NameSpec
    gen-ktcAPI.h, 221
cid_Next
    gen-ktcAPI.h, 221
cid_Params
    gen-ktcAPI.h, 221

cid_Placement
gen-ktcAPI.h, 221
cid_PropertyFuncs
gen-ktcAPI.h, 221
cid_Qualifier
gen-ktcAPI.h, 221
cid_Right
gen-ktcAPI.h, 221
cid_Size
gen-ktcAPI.h, 222
cid_Stmt
gen-ktcAPI.h, 222
cid_Stmts
gen-ktcAPI.h, 222
cid_TemplateName
gen-ktcAPI.h, 222
cid_TemplateParams
gen-ktcAPI.h, 222
cid_Then
gen-ktcAPI.h, 222
cid_Throw
gen-ktcAPI.h, 222
cid_TrailingReturnType
gen-ktcAPI.h, 222
cid_Type
gen-ktcAPI.h, 223
cid_TypeIds
gen-ktcAPI.h, 223
cid_Upper
gen-ktcAPI.h, 223

for defect, 16
Functions for accessing type information, 12

gen-ktcAPI.h, 41
cid_Adjacent, 215
cid_Args, 216
cid_Attributes, 216
cid_AttributeSpec, 216
cid_AttributeSpecs, 216
cid_Base, 216
cid_BaseSpecs, 216
cid_Bits, 216
cid_Cond, 216
cid_ConversionType, 217
cid_CtorInit, 217
cid_CVQualifiers, 217
cid_Decl, 217
cid_Declarator, 217
cid_Declarators, 217
cid_Decls, 217
cid_DeclSpecs, 217
cid_Default, 218
cid_Designators, 218
cid_Else, 218
cid_Enumerators, 218
cid_Exception, 218
cid_Expr, 218
cid_Func, 218
cid_FuncBody, 218
cid_Handler, 219
cid_Handlers, 219
cid_Index, 219
cid_Init, 219
cid_Initializer, 219
cid_Inits, 219
cid_Introducer, 219
cid_KRParams, 219
cid_Label, 220
cid_LambdaCapture, 220
cid_Left, 220
cid_Literal, 220
cid_Lower, 220
cid_MemberDecl, 220
cid_MemberDecls, 220
cid_MemberInitializers, 220
cid_Name, 221
cid_NameSpec, 221
cid_Next, 221
cid_Params, 221
cid_Placement, 221
cid_PropertyFuncs, 221
cid_Qualifier, 221
cid_Right, 221
cid_Size, 222
cid_Stmt, 222
cid_Stmts, 222
cid_TemplateName, 222
cid_TemplateParams, 222
cid_Then, 222
cid_Throw, 222
cid_TrailingReturnType, 222
cid_Type, 223
cid_TypeIds, 223
cid_Upper, 223
ktc_is_AccessSpecification, 53
ktc_is_AliasDecl, 53
ktc_is_AlignAsExpr, 54
ktc_is_AlignAsType, 54
ktc_is_AlignOfExpr, 55
ktc_is_AnyAttribute, 55
ktc_is_AnyCapture, 55
ktc_is_AnyDecl, 56
ktc_is_AnyDeclarator, 56
ktc_is_AnyDesignator, 56
ktc_is_AnyEnumerator, 57
ktc_is_AnyExpr, 57
ktc_is_AnyFuncBody, 57
ktc_is_AnyInitializer, 59
ktc_is_AnyLabel, 59
ktc_is_AnyMemberDecl, 59
ktc_is_AnyName, 61
ktc_is_AnyNameQualifier, 61
ktc_is_AnyNames, 61
ktc_is_AnyNameSpec, 63
ktc_is_AnyNonPtrDeclarator, 63
ktc_is_AnyParamName, 63

ktc_is_AnyPropertyFunc, 65
ktc_is_AnyPseudoDtor, 65
ktc_is_AnyStmt, 65
ktc_is_AnyTemplateArg, 67
ktc_is_AnyTypeName, 67
ktc_is_AnyTypeOf, 67
ktc_is_AnyTypeParam, 69
ktc_is_AnyUsing, 69
ktc_is_ArrayDeclarator, 69
ktc_is_AsmDef, 71
ktc_is_AsmStmt, 71
ktc_is_Attribute, 71
ktc_is_AttributedDeclarator, 73
ktc_is_AttributeDeclSpec, 73
ktc_is_Attributes, 73
ktc_is_AttributeSpec, 75
ktc_is_AttributeSpecs, 75
ktc_is_AttributeWithArgs, 75
ktc_is_AutoType, 77
ktc_is_BaseSpec, 77
ktc_is_BaseSpecs, 77
ktc_is_BinaryExpr, 79
ktc_is_BitFieldDeclarator, 79
ktc_is_BoolLiteralExpr, 79
ktc_is_BreakStmt, 81
ktc_is_BuiltinType, 81
ktc_is_CallExpr, 81
ktc_is_Capture, 83
ktc_is_CaptureDefault, 83
ktc_is_CaseLabel, 83
ktc_is_CaseRangeLabel, 85
ktc_is_CastExpr, 85
ktc_is_ClassType, 85
ktc_is_CompoundStmt, 87
ktc_is_ConditionalExpr, 87
ktc_is_ConstExpr, 87
ktc_is_ContinueStmt, 89
ktc_is_ConvFunc, 89
ktc_is_CopyInitializer, 89
ktc_is_CtorInitializer, 91
ktc_is_CVQualifier, 91
ktc_is_Decl, 91
ktc_is_DeclEllipsis, 93
ktc_is_DeclOrStmt, 93
ktc_is_DeclOrStmts, 93
ktc_is_DeclSpec, 95
ktc_is_DeclSpecs, 95
ktc_is_DefaultException, 95
ktc_is_DefaultLabel, 97
ktc_is_DeleteExpr, 97
ktc_is_DenyThrowSpec, 97
ktc_is_Designators, 99
ktc_is_DirectInitializer, 99
ktc_is_DoDeclStmt, 99
ktc_is_DoStmt, 101
ktc_is_Dtor, 101
ktc_is_Enumerator, 101
ktc_is_Enumerators, 103
ktc_is_EnumType, 103
ktc_is_ExceptHandler, 103
ktc_is_Exception, 105
ktc_is_ExceptionSpec, 105
ktc_is_ExplicitInstantiation, 105
ktc_is_ExprArg, 107
ktc_is_Exprs, 107
ktc_is_ExprStmt, 107
ktc_is_ExprTypeIdExpr, 109
ktc_is_FieldDesignator, 109
ktc_is_FinallyHandler, 109
ktc_is_ForEachStmt, 111
ktc_is_ForRangeStmt, 111
ktc_is_ForStmt, 111
ktc_is_FuncBody, 113
ktc_is_FuncDeclarator, 113
ktc_is_FuncDef, 113
ktc_is_FuncSpec, 115
ktc_is_FuncTryBlock, 115
ktc_is_GenericAttribute, 115
ktc_is_GlobalScope, 117
ktc_is_GotoStmt, 117
ktc_is_Handler, 117
ktc_is_Handlers, 119
ktc_is_IdExpr, 119
ktc_is_IfDeclStmt, 119
ktc_is_IfStmt, 121
ktc_is_IndexDesignator, 121
ktc_is_IndexExpr, 121
ktc_is_InitClause, 123
ktc_is_InitializedDeclarator, 123
ktc_is_InitializerExpr, 123
ktc_is_Initializers, 125
ktc_is_KRFuncDeclarator, 125
ktc_is_Label, 125
ktc_is_LabeledStmt, 127
ktc_is_LambdaDeclarator, 127
ktc_is_LambdaExpr, 127
ktc_is_LambdaIntroducer, 129
ktc_is_LeaveStmt, 129
ktc_is_LinkageSpec, 129
ktc_is_LiteralExpr, 131
ktc_is_MaybeCtorInitializer, 131
ktc_is_MaybeDeclarator, 131
ktc_is_MaybeException, 133
ktc_is_MaybeExceptionSpec, 133
ktc_is_MaybeLambdaDeclarator, 133
ktc_is_MaybeNewInitializer, 135
ktc_is_MaybeTypeId, 135
ktc_is_MemberDecl, 135
ktc_is_MemberDecls, 137
ktc_is_MemberDesignator, 137
ktc_is_MemberExpr, 137
ktc_is_MemberFunc, 139
ktc_is_MemberInitializer, 139
ktc_is_MemberInitializers, 139
ktc_is_MemberTemplate, 141
ktc_is_MemberUsingDecl, 141

ktc_is_Name, 141
ktc_is_NameDeclarator, 143
ktc_is_NamespaceAlias, 143
ktc_is_NamespaceDecl, 143
ktc_is_NameSpec, 145
ktc_is_NewExpr, 145
ktc_is_NewInitializer, 145
ktc_is_NoAttribute, 147
ktc_is_NoAttributeSpec, 147
ktc_is_NoBaseSpec, 147
ktc_is_NoCapture, 149
ktc_is_NoCtorInitializer, 149
ktc_is_Node, 149
ktc_is_NoDeclarator, 151
ktc_is_NoDeclOrStmt, 151
ktc_is_NoDeclSpec, 151
ktc_is_NoDesignator, 153
ktc_is_NoEnumerator, 153
ktc_is_NoException, 153
ktc_is_NoExceptionSpec, 155
ktc_is_NoExpr, 155
ktc_is_NoHandler, 155
ktc_is_NoInitializer, 157
ktc_is_NoLambdaDeclarator, 157
ktc_is_NoMemberDecl, 157
ktc_is_NoMemberInitializer, 159
ktc_is_NoName, 159
ktc_is_NoNameQualifier, 159
ktc_is_NoNewInitializer, 161
ktc_is_NoParamName, 161
ktc_is_NoPropertyFunc, 161
ktc_is_NoTemplateArg, 163
ktc_is_NoTemplateParam, 163
ktc_is_NoTypeId, 163
ktc_is_NullptrLiteralExpr, 165
ktc_is_OpFunc, 165
ktc_is_Param, 165
ktc_is_ParamName, 167
ktc_is_ParamNames, 167
ktc_is_ParensDeclarator, 167
ktc_is_ParensExpr, 169
ktc_is_PromisedFuncBody, 169
ktc_is_PromisedMemberDecl, 169
ktc_is_PropertyAttribute, 171
ktc_is_PropertyFuncs, 171
ktc_is_PropertyGetFunc, 171
ktc_is_PropertyPutFunc, 173
ktc_is_PseudoDtor, 173
ktc_is_PtrDeclarator, 173
ktc_is_QualifiedName, 175
ktc_is_QualifiedPseudoDtor, 175
ktc_is_RangeDesignator, 175
ktc_is_ReservedTypeSpec, 177
ktc_is_ReturnStmt, 177
ktc_is_SizeOfExpr, 177
ktc_is_SpecialCastExpr, 179
ktc_is_StaticAssertDecl, 179
ktc_is_StmtExpr, 179
ktc_is_StorageClass, 181
ktc_is_StringLiteralExpr, 181
ktc_is_SuffixFunc, 181
ktc_is_SuperScope, 183
ktc_is_SwitchDeclStmt, 183
ktc_is_SwitchStmt, 183
ktc_is_TemplateArgs, 185
ktc_is_TemplateDecl, 185
ktc_is_TemplateName, 185
ktc_is_TemplateParam, 187
ktc_is_TemplateParams, 187
ktc_is_TemplateSpec, 187
ktc_is_TemplateTypeArg, 189
ktc_is_TemplateTypeParam, 189
ktc_is_ThisExpr, 189
ktc_is_ThrowExpr, 191
ktc_is_TranslationUnit, 191
ktc_is_TruncatedInitClause, 191
ktc_is_TryExceptStmt, 193
ktc_is_TryFinallyStmt, 193
ktc_is_TryStmt, 193
ktc_is_TypeAdjective, 195
ktc_is_TypeArg, 195
ktc_is_TypeConvExpr, 195
ktc_is_TypeId, 197
ktc_is_TypeName, 197
ktc_is_TypeOfExpr, 197
ktc_is_TypeOfSpec, 199
ktc_is_TypeOfType, 199
ktc_is_TypeParam, 199
ktc_is_TypeTypeIdExpr, 201
ktc_is_UnaryExpr, 201
ktc_is_UnparsedDecl, 201
ktc_is_UnparsedDeclarator, 203
ktc_is_UnparsedDeclSpec, 203
ktc_is_UnparsedEnumerator, 203
ktc_is_UnparsedException, 205
ktc_is_UnparsedExpr, 205
ktc_is_UnparsedInitializer, 205
ktc_is_UnparsedLabel, 207
ktc_is_UnparsedMemberDecl, 207
ktc_is_UnparsedName, 207
ktc_is_UnparsedNameQualifier, 209
ktc_is_UnparsedParamName, 209
ktc_is_UnparsedPropertyFunc, 209
ktc_is_UnparsedStmt, 211
ktc_is_UnqualifiedName, 211
ktc_is_UserLiteralExpr, 211
ktc_is_UserStringLiteralExpr, 213
ktc_is_UsingDecl, 213
ktc_is_UsingDirective, 213
ktc_is_WhileDeclStmt, 215
ktc_is_WhileStmt, 215
tid_AccessSpecification, 223
tid_AliasDecl, 223
tid_AlignAsExpr, 223
tid_AlignAsType, 223
tid_AlignOfExpr, 223

tid_Any, 224
tid_AnyAttribute, 224
tid_AnyCapture, 224
tid_AnyDecl, 224
tid_AnyDeclarator, 224
tid_AnyDesignator, 224
tid_AnyEnumerator, 224
tid_AnyExpr, 224
tid_AnyFuncBody, 225
tid_AnyInitializer, 225
tid_AnyLabel, 225
tid_AnyMemberDecl, 225
tid_AnyName, 225
tid_AnyNameQualifier, 225
tid_AnyNames, 225
tid_AnyNameSpec, 225
tid_AnyNonPtrDeclarator, 226
tid_AnyParamName, 226
tid_AnyPropertyFunc, 226
tid_AnyPseudoDtor, 226
tid_AnyStmt, 226
tid_AnyTemplateArg, 226
tid_AnyTypeName, 226
tid_AnyTypeOf, 226
tid_AnyTypeParam, 227
tid_AnyUsing, 227
tid_ArrayDeclarator, 227
tid_AsmDef, 227
tid_AsmStmt, 227
tid_Attribute, 227
tid_AttributedDeclarator, 227
tid_AttributeDeclSpec, 227
tid_Attributes, 228
tid_AttributeSpec, 228
tid_AttributeSpecs, 228
tid_AttributeWithArgs, 228
tid_AutoType, 228
tid_BaseSpec, 228
tid_BaseSpecs, 228
tid_BinaryExpr, 228
tid_BitFieldDeclarator, 229
tid_BoolLiteralExpr, 229
tid_BreakStmt, 229
tid_BuiltinType, 229
tid_CallExpr, 229
tid_Capture, 229
tid_CaptureDefault, 229
tid_CaseLabel, 229
tid_CaseRangeLabel, 230
tid_CastExpr, 230
tid_ClassType, 230
tid_CompoundStmt, 230
tid_ConditionalExpr, 230
tid_ConstExpr, 230
tid_ContinueStmt, 230
tid_ConvFunc, 230
tid_CopyInitializer, 231
tid_CtorInitializer, 231
tid_CVQualifier, 231
tid_Decl, 231
tid_DeclEllipsis, 231
tid_DeclOrStmt, 231
tid_DeclOrStmts, 231
tid_DeclSpec, 231
tid_DeclSpecs, 232
tid_DefaultException, 232
tid_DefaultLabel, 232
tid_DeleteExpr, 232
tid_DenyThrowSpec, 232
tid_Designators, 232
tid_DirectInitializer, 232
tid_DoDeclStmt, 232
tid_DoStmt, 233
tid_Dtor, 233
tid_Enumerator, 233
tid_Enumerators, 233
tid_EnumType, 233
tid_ExceptHandler, 233
tid_Exception, 233
tid_ExceptionSpec, 233
tid_ExplicitInstantiation, 234
tid_ExprArg, 234
tid_Exprs, 234
tid_ExprStmt, 234
tid_ExprTypeIdExpr, 234
tid_FieldDesignator, 234
tid_FinallyHandler, 234
tid_ForEachStmt, 234
tid_ForRangeStmt, 235
tid_ForStmt, 235
tid_FuncBody, 235
tid_FuncDeclarator, 235
tid_FuncDef, 235
tid_FuncSpec, 235
tid_FuncTryBlock, 235
tid_GenericAttribute, 235
tid_GlobalScope, 236
tid_GotoStmt, 236
tid_Handler, 236
tid_Handlers, 236
tid_IdExpr, 236
tid_IfDeclStmt, 236
tid_IfStmt, 236
tid_IndexDesignator, 236
tid_IndexExpr, 237
tid_InitClause, 237
tid_InitializedDeclarator, 237
tid_InitializerExpr, 237
tid_Initializers, 237
tid_KRFuncDeclarator, 237
tid_Label, 237
tid_LabeledStmt, 237
tid_LambdaDeclarator, 238
tid_LambdaExpr, 238
tid_LambdaIntroducer, 238
tid_LeaveStmt, 238

tid_LinkageSpec, 238
tid_LiteralExpr, 238
tid_MaybeCtorInitializer, 238
tid_MaybeDeclarator, 238
tid_MaybeException, 239
tid_MaybeExceptionSpec, 239
tid_MaybeLambdaDeclarator, 239
tid_MaybeNewInitializer, 239
tid_MaybeTypeId, 239
tid_MemberDecl, 239
tid_MemberDecls, 239
tid_MemberDesignator, 239
tid_MemberExpr, 240
tid_MemberFunc, 240
tid_MemberInitializer, 240
tid_MemberInitializers, 240
tid_MemberTemplate, 240
tid_MemberUsingDecl, 240
tid_Name, 240
tid_NameDeclarator, 240
tid_NamespaceAlias, 241
tid_NamespaceDecl, 241
tid_NameSpec, 241
tid_NewExpr, 241
tid_NewInitializer, 241
tid_NoAttribute, 241
tid_NoAttributeSpec, 241
tid_NoBaseSpec, 241
tid_NoCapture, 242
tid_NoCtorInitializer, 242
tid_Node, 242
tid_NoDeclarator, 242
tid_NoDeclOrStmt, 242
tid_NoDeclSpec, 242
tid_NoDesignator, 242
tid_NoEnumerator, 242
tid_NoException, 243
tid_NoExceptionSpec, 243
tid_NoExpr, 243
tid_NoHandler, 243
tid_NoInitializer, 243
tid_NoLambdaDeclarator, 243
tid_NoMemberDecl, 243
tid_NoMemberInitializer, 243
tid_NoName, 244
tid_NoNameQualifier, 244
tid_NoNewInitializer, 244
tid_NoParamName, 244
tid_NoPropertyFunc, 244
tid_NoTemplateArg, 244
tid_NoTemplateParam, 244
tid_NoTypeId, 244
tid_NullptrLiteralExpr, 245
tid_OpFunc, 245
tid_Param, 245
tid_ParamName, 245
tid_ParamNames, 245
tid_ParensDeclarator, 245
tid_ParensExpr, 245
tid_PromisedFuncBody, 245
tid_PromisedMemberDecl, 246
tid_PropertyAttribute, 246
tid_PropertyFuncs, 246
tid_PropertyGetFunc, 246
tid_PropertyPutFunc, 246
tid_PseudoDtor, 246
tid_PtrDeclarator, 246
tid_QualifiedName, 246
tid_QualifiedPseudoDtor, 247
tid_RangeDesignator, 247
tid_ReservedTypeSpec, 247
tid_ReturnStmt, 247
tid_SizeOfExpr, 247
tid_SpecialCastExpr, 247
tid_StaticAssertDecl, 247
tid_StmtExpr, 247
tid_StorageClass, 248
tid_StringLiteralExpr, 248
tid_SuffixFunc, 248
tid_SuperScope, 248
tid_SwitchDeclStmt, 248
tid_SwitchStmt, 248
tid_TemplateArgs, 248
tid_TemplateDecl, 248
tid_TemplateName, 249
tid_TemplateParam, 249
tid_TemplateParams, 249
tid_TemplateSpec, 249
tid_TemplateTypeArg, 249
tid_TemplateTypeParam, 249
tid_ThisExpr, 249
tid_ThrowExpr, 249
tid_TranslationUnit, 250
tid_TruncatedInitClause, 250
tid_TryExceptStmt, 250
tid_TryFinallyStmt, 250
tid_TryStmt, 250
tid_TypeAdjective, 250
tid_TypeArg, 250
tid_TypeConvExpr, 250
tid_TypeId, 251
tid_TypeName, 251
tid_TypeOfExpr, 251
tid_TypeOfSpec, 251
tid_TypeOfType, 251
tid_TypeParam, 251
tid_TypeTypeIdExpr, 251
tid_UnaryExpr, 251
tid_UnparsedDecl, 252
tid_UnparsedDeclarator, 252
tid_UnparsedDeclSpec, 252
tid_UnparsedEnumerator, 252
tid_UnparsedException, 252
tid_UnparsedExpr, 252
tid_UnparsedInitializer, 252
tid_UnparsedLabel, 252

tid_UnparsedMemberDecl, 253
tid_UnparsedName, 253
tid_UnparsedNameQualifier, 253
tid_UnparsedParamName, 253
tid_UnparsedPropertyFunc, 253
tid_UnparsedStmt, 253
tid_UnqualifiedName, 253
tid_UserLiteralExpr, 253
tid_UserStringLiteralExpr, 254
tid_UsingDecl, 254
tid_UsingDirective, 254
tid_WhileDeclStmt, 254
tid_WhileStmt, 254

KTC_API_VERSION_MAJOR
ktcMainAPI.h, 260
KTC_API_VERSION_MINOR
ktcMainAPI.h, 260
KTC_API_VERSION_PATCHLEVEL
ktcMainAPI.h, 260
ktc_assembleStringConstant
ktcMainAPI.h, 262
ktc_autofix_addSegment
ktcMainAPI.h, 264
ktc_autofix_delete
ktcMainAPI.h, 264
ktc_autofix_new
ktcMainAPI.h, 264
ktc_autofix_t
ktcMainAPI.h, 260
KTC_BUILTINTYPE_BOOL
Numerical codes of C/C++ built-in types, 34
KTC_BUILTINTYPE_CHAR
Numerical codes of C/C++ built-in types, 34
KTC_BUILTINTYPE_DOUBLE
Numerical codes of C/C++ built-in types, 34
KTC_BUILTINTYPE_FLOAT
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_INT
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_LONGDOUBLE
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_LONGINT
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_LONGLONGINT
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_NONE
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_NULLPTR_T
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_SHORTINT
Numerical codes of C/C++ built-in types, 35
KTC_BUILTINTYPE_SIGNEDCHAR
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_SIGNEDINT
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_SIGNEDLONGINT
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_SIGNEDLONGLONGINT
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_SIGNEDSHORTINT
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_UNSIGNEDCHAR
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_UNSIGNEDINT
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_UNSIGNEDLONGINT
Numerical codes of C/C++ built-in types, 36
KTC_BUILTINTYPE_UNSIGNEDLONGLONGINT
Numerical codes of C/C++ built-in types, 37
KTC_BUILTINTYPE_UNSIGNEDSHORTINT
Numerical codes of C/C++ built-in types, 37
KTC_BUILTINTYPE_VOID
Numerical codes of C/C++ built-in types, 37
KTC_BUILTINTYPE_WCHAR_T
Numerical codes of C/C++ built-in types, 37
KTC_CASTSPECIFIER_CONST
Numerical codes for identifying different C++ -style cast expressions, 39
KTC_CASTSPECIFIER_DYNAMIC
Numerical codes for identifying different C++ -style cast expressions, 39
KTC_CASTSPECIFIER_REINTERPRET
Numerical codes for identifying different C++ -style cast expressions, 39
KTC_CASTSPECIFIER_STATIC
Numerical codes for identifying different C++ -style cast expressions, 39
ktc_childId_t
ktcMainAPI.h, 261
KTC_CLASSTAG_CLASS
Numerical codes to differentiate struct/class/union declarations, 33
KTC_CLASSTAG_NONE
Numerical codes to differentiate struct/class/union declarations, 33
KTC_CLASSTAG_STRUCT
Numerical codes to differentiate struct/class/union declarations, 33
KTC_CLASSTAG_UNION
Numerical codes to differentiate struct/class/union declarations, 33
ktc_compareSubtrees
ktcMainAPI.h, 264
ktc_constructorName
ktcMainAPI.h, 318
KTC_CUSTOM_TYPES
ktcMainAPI.h, 260
KTC_CVQUALIFIER_CONST
Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values, 32
KTC_CVQUALIFIER_NONE
Numerical codes of declaration type quali-

fiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values, 32
KTC_CVQUALIFIER_RESTRICT
Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values, 32
KTC_CVQUALIFIER_VOLATILE
Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values, 32
ktc_destructorName
ktcMainAPI.h, 319
KTC_DIALECT_ARM
ktcMainAPI.h, 319
KTC_DIALECT_GHS
ktcMainAPI.h, 319
KTC_DIALECT_GNU
ktcMainAPI.h, 319
KTC_DIALECT_MS
ktcMainAPI.h, 319
KTC_DIALECT_STD
ktcMainAPI.h, 319
KTC_DIALECT_SUN
ktcMainAPI.h, 319
KTC_DIALECT_TI
ktcMainAPI.h, 319
ktc_error_getConfigurationParameter
ktcMainAPI.h, 265
kwapi.h, 324
ktc_error_isEnabled
ktcMainAPI.h, 265
ktc_event_new
ktcMainAPI.h, 265
ktc_event_setParameter
ktcMainAPI.h, 266
ktc_eventHook_t
ktcMainAPI.h, 261
ktc_forAllSubtreeNodes
ktcMainAPI.h, 266
ktc_free
ktcMainAPI.h, 266
KTC_FUNCSPECIFIER_EXPLICIT
Numerical codes for identifying inline/virtual/explicit/friend member function specifiers, 38
KTC_FUNCSPECIFIER_FRIEND
Numerical codes for identifying inline/virtual/explicit/friend member function specifiers, 38
KTC_FUNCSPECIFIER_INLINE
Numerical codes for identifying inline/virtual/explicit/friend member function specifiers, 38
KTC_FUNCSPECIFIER_NONE
Numerical codes for identifying inline/virtual/explicit/friend member function specifiers, 38
KTC_FUNCSPECIFIER_VIRTUAL
Numerical codes for identifying inline/virtual/explicit/friend member function specifiers, 38
ktc_get_child_index
Numerical codes for identifying declarator ptr types, 40
ktc_getAssociatedScope
ktcMainAPI.h, 266
ktc_getBuiltinType
ktcMainAPI.h, 267
ktc_getBuiltinTypeSize
ktcMainAPI.h, 267
ktc_getCallArgument
ktcMainAPI.h, 268
ktc_getCalledFunction
ktcMainAPI.h, 268
ktc_getCastSpecifier
ktcMainAPI.h, 268
ktc_getClassTag
ktcMainAPI.h, 269
ktc_getEndPosition
ktcMainAPI.h, 269
ktc_getFrontendDialect
ktcMainAPI.h, 269
ktc_getFrontendLanguage
ktcMainAPI.h, 269
ktc_getFunctionSpecifier
ktcMainAPI.h, 270
ktc_getIdentifier
ktcMainAPI.h, 270
ktc_getIdentifierNo
ktcMainAPI.h, 270
ktc_getIntegerValue
ktcMainAPI.h, 270
ktc_getLanguageType
ktcMainAPI.h, 271
ktc_getNameDeclarator
ktcMainAPI.h, 271
ktc_getNoIdent
ktcMainAPI.h, 271
ktc_getNumberOfCallArguments
ktcMainAPI.h, 271
ktc_getOperation
ktcMainAPI.h, 272
ktc_getOutputFileName
ktcMainAPI.h, 272
ktc_getPointedType
ktcMainAPI.h, 272
ktc_getPointerOperator
ktcMainAPI.h, 272
ktc_getPointerSize
ktcMainAPI.h, 273
ktc_getSemanticInfo
ktcMainAPI.h, 273

ktc_getSizeofArgument
ktcMainAPI.h, 273
ktc_getStartPosition
ktcMainAPI.h, 273
ktc_getStorageClass
ktcMainAPI.h, 274
ktc_getStringConstantValue
ktcMainAPI.h, 274
ktc_getTokens
ktcMainAPI.h, 274
ktc_getTypeQualifiers
ktcMainAPI.h, 274
ktc_hasBuiltinWideChar
ktcMainAPI.h, 275
ktc_is_AccessSpecification
gen-ktcAPI.h, 53
ktc_is_AliasDecl
gen-ktcAPI.h, 53
ktc_is_AlignAsExpr
gen-ktcAPI.h, 54
ktc_is_AlignAsType
gen-ktcAPI.h, 54
ktc_is_AlignOfExpr
gen-ktcAPI.h, 55
ktc_is_AnyAttribute
gen-ktcAPI.h, 55
ktc_is_AnyCapture
gen-ktcAPI.h, 55
ktc_is_AnyDecl
gen-ktcAPI.h, 56
ktc_is_AnyDeclarator
gen-ktcAPI.h, 56
ktc_is_AnyDesignator
gen-ktcAPI.h, 56
ktc_is_AnyEnumerator
gen-ktcAPI.h, 57
ktc_is_AnyExpr
gen-ktcAPI.h, 57
ktc_is_AnyFuncBody
gen-ktcAPI.h, 57
ktc_is_AnyInitializer
gen-ktcAPI.h, 59
ktc_is_AnyLabel
gen-ktcAPI.h, 59
ktc_is_AnyMemberDecl
gen-ktcAPI.h, 59
ktc_is_AnyName
gen-ktcAPI.h, 61
ktc_is_AnyNameQualifier
gen-ktcAPI.h, 61
ktc_is_AnyNames
gen-ktcAPI.h, 61
ktc_is_AnyNameSpec
gen-ktcAPI.h, 63
ktc_is_AnyNonPtrDeclarator
gen-ktcAPI.h, 63
ktc_is_AnyParamName
gen-ktcAPI.h, 63
ktc_is_AnyPropertyFunc
gen-ktcAPI.h, 65
ktc_is_AnyPseudoDtor
gen-ktcAPI.h, 65
ktc_is_AnyStmt
gen-ktcAPI.h, 65
ktc_is_AnyTemplateArg
gen-ktcAPI.h, 67
ktc_is_AnyTypeName
gen-ktcAPI.h, 67
ktc_is_AnyTypeOf
gen-ktcAPI.h, 67
ktc_is_AnyTypeParam
gen-ktcAPI.h, 69
ktc_is_AnyUsing
gen-ktcAPI.h, 69
ktc_is_ArrayDeclarator
gen-ktcAPI.h, 69
ktc_is_AsmDef
gen-ktcAPI.h, 71
ktc_is_AsmStmt
gen-ktcAPI.h, 71
ktc_is_Attribute
gen-ktcAPI.h, 71
ktc_is_AttributedDeclarator
gen-ktcAPI.h, 73
ktc_is_AttributeDeclSpec
gen-ktcAPI.h, 73
ktc_is_Attributes
gen-ktcAPI.h, 73
ktc_is_AttributeSpec
gen-ktcAPI.h, 75
ktc_is_AttributeSpecs
gen-ktcAPI.h, 75
ktc_is_AttributeWithArgs
gen-ktcAPI.h, 75
ktc_is_AutoType
gen-ktcAPI.h, 77
ktc_is_BaseSpec
gen-ktcAPI.h, 77
ktc_is_BaseSpecs
gen-ktcAPI.h, 77
ktc_is_BinaryExpr
gen-ktcAPI.h, 79
ktc_is_BitFieldDeclarator
gen-ktcAPI.h, 79
ktc_is_BoolLiteralExpr
gen-ktcAPI.h, 79
ktc_is_BreakStmt
gen-ktcAPI.h, 81
ktc_is_BuiltinType
gen-ktcAPI.h, 81
ktc_is_CallExpr
gen-ktcAPI.h, 81
ktc_is_Capture
gen-ktcAPI.h, 83
ktc_is_CaptureDefault
gen-ktcAPI.h, 83

ktc_is_CaseLabel
gen-ktcAPI.h, 83
ktc_is_CaseRangeLabel
gen-ktcAPI.h, 85
ktc_is_CastExpr
gen-ktcAPI.h, 85
ktc_is_ClassType
gen-ktcAPI.h, 85
ktc_is_CompoundStmt
gen-ktcAPI.h, 87
ktc_is_ConditionalExpr
gen-ktcAPI.h, 87
ktc_is_ConstExpr
gen-ktcAPI.h, 87
ktc_is_ContinueStmt
gen-ktcAPI.h, 89
ktc_is_ConvFunc
gen-ktcAPI.h, 89
ktc_is_CopyInitializer
gen-ktcAPI.h, 89
ktc_is_CtorInitializer
gen-ktcAPI.h, 91
ktc_is_CVQualifier
gen-ktcAPI.h, 91
ktc_is_Decl
gen-ktcAPI.h, 91
ktc_is_DeclEllipsis
gen-ktcAPI.h, 93
ktc_is_DeclOrStmt
gen-ktcAPI.h, 93
ktc_is_DeclOrStmts
gen-ktcAPI.h, 93
ktc_is_DeclSpec
gen-ktcAPI.h, 95
ktc_is_DeclSpecs
gen-ktcAPI.h, 95
ktc_is_DefaultException
gen-ktcAPI.h, 95
ktc_is_DefaultLabel
gen-ktcAPI.h, 97
ktc_is_DeleteExpr
gen-ktcAPI.h, 97
ktc_is_DenyThrowSpec
gen-ktcAPI.h, 97
ktc_is_Designators
gen-ktcAPI.h, 99
ktc_is_DirectInitializer
gen-ktcAPI.h, 99
ktc_is_DoDeclStmt
gen-ktcAPI.h, 99
ktc_is_DoStmt
gen-ktcAPI.h, 101
ktc_is_Dtor
gen-ktcAPI.h, 101
ktc_is_Enumerator
gen-ktcAPI.h, 101
ktc_is_Enumerators
gen-ktcAPI.h, 103
ktc_is_EnumType
gen-ktcAPI.h, 103
ktc_is_ExceptHandler
gen-ktcAPI.h, 103
ktc_is_Exception
gen-ktcAPI.h, 105
ktc_is_ExceptionSpec
gen-ktcAPI.h, 105
ktc_is_ExplicitInstantiation
gen-ktcAPI.h, 105
ktc_is_ExprArg
gen-ktcAPI.h, 107
ktc_is_Exprs
gen-ktcAPI.h, 107
ktc_is_ExprStmt
gen-ktcAPI.h, 107
ktc_is_ExprTypeIdExpr
gen-ktcAPI.h, 109
ktc_is_FieldDesignator
gen-ktcAPI.h, 109
ktc_is_FinallyHandler
gen-ktcAPI.h, 109
ktc_is_ForEachStmt
gen-ktcAPI.h, 111
ktc_is_ForRangeStmt
gen-ktcAPI.h, 111
ktc_is_ForStmt
gen-ktcAPI.h, 111
ktc_is_FuncBody
gen-ktcAPI.h, 113
ktc_is_FuncDeclarator
gen-ktcAPI.h, 113
ktc_is_FuncDef
gen-ktcAPI.h, 113
ktc_is_FuncSpec
gen-ktcAPI.h, 115
ktc_is_FuncTryBlock
gen-ktcAPI.h, 115
ktc_is_GenericAttribute
gen-ktcAPI.h, 115
ktc_is_GlobalScope
gen-ktcAPI.h, 117
ktc_is_GotoStmt
gen-ktcAPI.h, 117
ktc_is_Handler
gen-ktcAPI.h, 117
ktc_is_Handlers
gen-ktcAPI.h, 119
ktc_is_IdExpr
gen-ktcAPI.h, 119
ktc_is_IfDeclStmt
gen-ktcAPI.h, 119
ktc_is_IfStmt
gen-ktcAPI.h, 121
ktc_is_IndexDesignator
gen-ktcAPI.h, 121
ktc_is_IndexExpr
gen-ktcAPI.h, 121

ktc_is_InitClause
gen-ktcAPI.h, 123
ktc_is_InitializedDeclarator
gen-ktcAPI.h, 123
ktc_is_InitializerExpr
gen-ktcAPI.h, 123
ktc_is_Initializers
gen-ktcAPI.h, 125
ktc_is_KRFuncDeclarator
gen-ktcAPI.h, 125
ktc_is_Label
gen-ktcAPI.h, 125
ktc_is_LabeledStmt
gen-ktcAPI.h, 127
ktc_is_LambdaDeclarator
gen-ktcAPI.h, 127
ktc_is_LambdaExpr
gen-ktcAPI.h, 127
ktc_is_LambdaIntroducer
gen-ktcAPI.h, 129
ktc_is_LeaveStmt
gen-ktcAPI.h, 129
ktc_is_LinkageSpec
gen-ktcAPI.h, 129
ktc_is_LiteralExpr
gen-ktcAPI.h, 131
ktc_is_MaybeCtorInitializer
gen-ktcAPI.h, 131
ktc_is_MaybeDeclarator
gen-ktcAPI.h, 131
ktc_is_MaybeException
gen-ktcAPI.h, 133
ktc_is_MaybeExceptionSpec
gen-ktcAPI.h, 133
ktc_is_MaybeLambdaDeclarator
gen-ktcAPI.h, 133
ktc_is_MaybeNewInitializer
gen-ktcAPI.h, 135
ktc_is_MaybeTypeId
gen-ktcAPI.h, 135
ktc_is_MemberDecl
gen-ktcAPI.h, 135
ktc_is_MemberDecls
gen-ktcAPI.h, 137
ktc_is_MemberDesignator
gen-ktcAPI.h, 137
ktc_is_MemberExpr
gen-ktcAPI.h, 137
ktc_is_MemberFunc
gen-ktcAPI.h, 139
ktc_is_MemberInitializer
gen-ktcAPI.h, 139
ktc_is_MemberInitializers
gen-ktcAPI.h, 139
ktc_is_MemberTemplate
gen-ktcAPI.h, 141
ktc_is_MemberUsingDecl
gen-ktcAPI.h, 141
ktc_is_Name
gen-ktcAPI.h, 141
ktc_is_NameDeclarator
gen-ktcAPI.h, 143
ktc_is_NamespaceAlias
gen-ktcAPI.h, 143
ktc_is_NamespaceDecl
gen-ktcAPI.h, 143
ktc_is_NameSpec
gen-ktcAPI.h, 145
ktc_is_NewExpr
gen-ktcAPI.h, 145
ktc_is_NewInitializer
gen-ktcAPI.h, 145
ktc_is_NoAttribute
gen-ktcAPI.h, 147
ktc_is_NoAttributeSpec
gen-ktcAPI.h, 147
ktc_is_NoBaseSpec
gen-ktcAPI.h, 147
ktc_is_NoCapture
gen-ktcAPI.h, 149
ktc_is_NoCtorInitializer
gen-ktcAPI.h, 149
ktc_is_Node
gen-ktcAPI.h, 149
ktc_is_NoDeclarator
gen-ktcAPI.h, 151
ktc_is_NoDeclOrStmt
gen-ktcAPI.h, 151
ktc_is_NoDeclSpec
gen-ktcAPI.h, 151
ktc_is_NoDesignator
gen-ktcAPI.h, 153
ktc_is_NoEnumerator
gen-ktcAPI.h, 153
ktc_is_NoException
gen-ktcAPI.h, 153
ktc_is_NoExceptionSpec
gen-ktcAPI.h, 155
ktc_is_NoExpr
gen-ktcAPI.h, 155
ktc_is_NoHandler
gen-ktcAPI.h, 155
ktc_is_NoInitializer
gen-ktcAPI.h, 157
ktc_is_NoLambdaDeclarator
gen-ktcAPI.h, 157
ktc_is_NoMemberDecl
gen-ktcAPI.h, 157
ktc_is_NoMemberInitializer
gen-ktcAPI.h, 159
ktc_is_NoName
gen-ktcAPI.h, 159
ktc_is_NoNameQualifier
gen-ktcAPI.h, 159
ktc_is_NoNewInitializer
gen-ktcAPI.h, 161

ktc_is_NoParamName
gen-ktcAPI.h, 161
ktc_is_NoPropertyFunc
gen-ktcAPI.h, 161
ktc_is_NoTemplateArg
gen-ktcAPI.h, 163
ktc_is_NoTemplateParam
gen-ktcAPI.h, 163
ktc_is_NoToken
ktcMainAPI.h, 275
ktc_is_NoTypeId
gen-ktcAPI.h, 163
ktc_is_NullptrLiteralExpr
gen-ktcAPI.h, 165
ktc_is_OpFunc
gen-ktcAPI.h, 165
ktc_is_Param
gen-ktcAPI.h, 165
ktc_is_ParamName
gen-ktcAPI.h, 167
ktc_is_ParamNames
gen-ktcAPI.h, 167
ktc_is_ParensDeclarator
gen-ktcAPI.h, 167
ktc_is_ParensExpr
gen-ktcAPI.h, 169
ktc_is_PromisedFuncBody
gen-ktcAPI.h, 169
ktc_is_PromisedMemberDecl
gen-ktcAPI.h, 169
ktc_is_PropertyAttribute
gen-ktcAPI.h, 171
ktc_is_PropertyFuncs
gen-ktcAPI.h, 171
ktc_is_PropertyGetFunc
gen-ktcAPI.h, 171
ktc_is_PropertyPutFunc
gen-ktcAPI.h, 173
ktc_is_PseudoDtor
gen-ktcAPI.h, 173
ktc_is_PtrDeclarator
gen-ktcAPI.h, 173
ktc_is_QualifiedName
gen-ktcAPI.h, 175
ktc_is_QualifiedPseudoDtor
gen-ktcAPI.h, 175
ktc_is_RangeDesignator
gen-ktcAPI.h, 175
ktc_is_ReservedTypeSpec
gen-ktcAPI.h, 177
ktc_is_ReturnStmt
gen-ktcAPI.h, 177
ktc_is_SizeOfExpr
gen-ktcAPI.h, 177
ktc_is_SpecialCastExpr
gen-ktcAPI.h, 179
ktc_is_StaticAssertDecl
gen-ktcAPI.h, 179
ktc_is_StmtExpr
gen-ktcAPI.h, 179
ktc_is_StorageClass
gen-ktcAPI.h, 181
ktc_is_StringLiteralExpr
gen-ktcAPI.h, 181
ktc_is_SuffixFunc
gen-ktcAPI.h, 181
ktc_is_SuperScope
gen-ktcAPI.h, 183
ktc_is_SwitchDeclStmt
gen-ktcAPI.h, 183
ktc_is_SwitchStmt
gen-ktcAPI.h, 183
ktc_is_TemplateArgs
gen-ktcAPI.h, 185
ktc_is_TemplateDecl
gen-ktcAPI.h, 185
ktc_is_TemplateName
gen-ktcAPI.h, 185
ktc_is_TemplateParam
gen-ktcAPI.h, 187
ktc_is_TemplateParams
gen-ktcAPI.h, 187
ktc_is_TemplateSpec
gen-ktcAPI.h, 187
ktc_is_TemplateTypeArg
gen-ktcAPI.h, 189
ktc_is_TemplateTypeParam
gen-ktcAPI.h, 189
ktc_is_ThisExpr
gen-ktcAPI.h, 189
ktc_is_ThrowExpr
gen-ktcAPI.h, 191
ktc_is_Token
ktcMainAPI.h, 275
ktc_is_TranslationUnit
gen-ktcAPI.h, 191
ktc_is_TruncatedInitClause
gen-ktcAPI.h, 191
ktc_is_TryExceptStmt
gen-ktcAPI.h, 193
ktc_is_TryFinallyStmt
gen-ktcAPI.h, 193
ktc_is_TryStmt
gen-ktcAPI.h, 193
ktc_is_TypeAdjective
gen-ktcAPI.h, 195
ktc_is_TypeArg
gen-ktcAPI.h, 195
ktc_is_TypeConvExpr
gen-ktcAPI.h, 195
ktc_is_TypeId
gen-ktcAPI.h, 197
ktc_is_TypeName
gen-ktcAPI.h, 197
ktc_is_TypeOfExpr
gen-ktcAPI.h, 197

ktc_is_TypeOfSpec
gen-ktcAPI.h, 199
ktc_is_TypeOfType
gen-ktcAPI.h, 199
ktc_is_TypeParam
gen-ktcAPI.h, 199
ktc_is_TypeTypeIdExpr
gen-ktcAPI.h, 201
ktc_is_UnaryExpr
gen-ktcAPI.h, 201
ktc_is_UnparsedDecl
gen-ktcAPI.h, 201
ktc_is_UnparsedDeclarator
gen-ktcAPI.h, 203
ktc_is_UnparsedDeclSpec
gen-ktcAPI.h, 203
ktc_is_UnparsedEnumerator
gen-ktcAPI.h, 203
ktc_is_UnparsedException
gen-ktcAPI.h, 205
ktc_is_UnparsedExpr
gen-ktcAPI.h, 205
ktc_is_UnparsedInitializer
gen-ktcAPI.h, 205
ktc_is_UnparsedLabel
gen-ktcAPI.h, 207
ktc_is_UnparsedMemberDecl
gen-ktcAPI.h, 207
ktc_is_UnparsedName
gen-ktcAPI.h, 207
ktc_is_UnparsedNameQualifier
gen-ktcAPI.h, 209
ktc_is_UnparsedParamName
gen-ktcAPI.h, 209
ktc_is_UnparsedPropertyFunc
gen-ktcAPI.h, 209
ktc_is_UnparsedStmt
gen-ktcAPI.h, 211
ktc_is_UnqualifiedName
gen-ktcAPI.h, 211
ktc_is_UserLiteralExpr
gen-ktcAPI.h, 211
ktc_is_UserStringLiteralExpr
gen-ktcAPI.h, 213
ktc_is_UsingDecl
gen-ktcAPI.h, 213
ktc_is_UsingDirective
gen-ktcAPI.h, 213
ktc_is_WhileDeclStmt
gen-ktcAPI.h, 215
ktc_is_WhileStmt
gen-ktcAPI.h, 215
ktc_isCallTo
ktcMainAPI.h, 276
ktc_isCharLiteral
ktcMainAPI.h, 276
ktc_isConstructor
ktcMainAPI.h, 276
ktc_isDeclaration
ktcMainAPI.h, 277
ktc_isDefinition
ktcMainAPI.h, 277
ktc_isIncluded
ktcMainAPI.h, 277
ktc_isMacroExpansion
ktcMainAPI.h, 277
ktc_isMacroExpansion2
ktcMainAPI.h, 278
ktc_isNameIncluded
ktcMainAPI.h, 278
ktc_isNullMacro
ktcMainAPI.h, 278
ktc_isNullPointerConstant
ktcMainAPI.h, 278
ktc_isOperationOverloaded
ktcMainAPI.h, 279
ktc_isStatic
ktcMainAPI.h, 279
ktc_isTreeType
ktcMainAPI.h, 279
ktc_isUTF16String
ktcMainAPI.h, 281
ktc_isUTF32String
ktcMainAPI.h, 281
ktc_isWideString
ktcMainAPI.h, 281
KTC_LANGUAGE_C
ktcMainAPI.h, 320
KTC_LANGUAGE_CXX
ktcMainAPI.h, 320
ktc_languageType_t
ktcMainAPI.h, 261
ktc_languageTypeEffectiveSignedness
ktcMainAPI.h, 282
ktc_languageTypeIsBuiltin
ktcMainAPI.h, 282
ktc_languageTypeIsPointer
ktcMainAPI.h, 282
ktc_languageTypeSignedness
ktcMainAPI.h, 282
ktc_languageTypeSize
ktcMainAPI.h, 283
ktc_long_long_t
ktcMainAPI.h, 261
ktc_message
ktcMainAPI.h, 261
ktc_message_addAnchorAttribute
ktcMainAPI.h, 283
ktc_message_addAttribute
ktcMainAPI.h, 283
ktc_message_addEvent
ktcMainAPI.h, 284
ktc_message_addEventEx
ktcMainAPI.h, 284
ktc_message_addTraceBySemanticsInfo
ktcMainAPI.h, 284

ktc_message_delete
ktcMainAPI.h, 284
ktc_message_new
ktcMainAPI.h, 284
ktc_message_render
ktcMainAPI.h, 285
ktc_message_render_wi_autofix
ktcMainAPI.h, 285
ktc_message_setFunction
ktcMainAPI.h, 285
ktc_message_setPosition
ktcMainAPI.h, 285
ktc_message_setRecommendationFactor
ktcMainAPI.h, 285
ktc_message_t
ktcMainAPI.h, 261
ktc_message_unsetFunction
ktcMainAPI.h, 286
ktc_nodeStackGet
ktcMainAPI.h, 286
ktc_nodeStackTop
ktcMainAPI.h, 286
KTC_OPCODE_ADD
Numerical codes of operations, 23
KTC_OPCODE_ADDASSIGN
Numerical codes of operations, 23
KTC_OPCODE_ADDRESS
Numerical codes of operations, 23
KTC_OPCODE_ANDASSIGN
Numerical codes of operations, 23
KTC_OPCODE_ASL
Numerical codes of operations, 23
KTC_OPCODE_ASLASSIGN
Numerical codes of operations, 23
KTC_OPCODE_ASR
Numerical codes of operations, 24
KTC_OPCODE_ASRASSIGN
Numerical codes of operations, 24
KTC_OPCODE_ASSIGN
Numerical codes of operations, 24
KTC_OPCODE_BITAND
Numerical codes of operations, 24
KTC_OPCODE_BITNOT
Numerical codes of operations, 24
KTC_OPCODE_BITOR
Numerical codes of operations, 24
KTC_OPCODE_BITXOR
Numerical codes of operations, 24
KTC_OPCODE_COMMA
Numerical codes of operations, 24
KTC_OPCODE_COND
Numerical codes of operations, 25
KTC_OPCODE_DEREF
Numerical codes of operations, 25
KTC_OPCODE_DEREFAST
Numerical codes of operations, 25
KTC_OPCODE_DIV
Numerical codes of operations, 25
KTC_OPCODE_DIVASSIGN
Numerical codes of operations, 25
KTC_OPCODE_DOTAST
Numerical codes of operations, 25
KTC_OPCODE_EQ
Numerical codes of operations, 25
KTC_OPCODE_FIELD
Numerical codes of operations, 25
KTC_OPCODE_FIELDREF
Numerical codes of operations, 26
KTC_OPCODE_GE
Numerical codes of operations, 26
KTC_OPCODE_GT
Numerical codes of operations, 26
KTC_OPCODE_LE
Numerical codes of operations, 26
KTC_OPCODE_LOGAND
Numerical codes of operations, 26
KTC_OPCODE_LOGNOT
Numerical codes of operations, 26
KTC_OPCODE_LOGOR
Numerical codes of operations, 26
KTC_OPCODE_LT
Numerical codes of operations, 26
KTC_OPCODE_MAX
Numerical codes of operations, 27
KTC_OPCODE_MIN
Numerical codes of operations, 27
KTC_OPCODE_MINUS
Numerical codes of operations, 27
KTC_OPCODE_MOD
Numerical codes of operations, 27
KTC_OPCODE_MODASSIGN
Numerical codes of operations, 27
KTC_OPCODE_MUL
Numerical codes of operations, 27
KTC_OPCODE_MULASSIGN
Numerical codes of operations, 27
KTC_OPCODE_NE
Numerical codes of operations, 27
KTC_OPCODE_NONE
Numerical codes of operations, 28
KTC_OPCODE_ORASSIGN
Numerical codes of operations, 28
KTC_OPCODE_PLUS
Numerical codes of operations, 28
KTC_OPCODE_POSTDEC
Numerical codes of operations, 28
KTC_OPCODE_POSTINC
Numerical codes of operations, 28
KTC_OPCODE_PREDEC
Numerical codes of operations, 28
KTC_OPCODE_PREINC
Numerical codes of operations, 28
KTC_OPCODE_ROUND_BRACKETS
Numerical codes of operations, 28
KTC_OPCODE_SIZEOF
Numerical codes of operations, 29

KTC_OPCODE_SQUARE_BRACKETS
Numerical codes of operations, 29
KTC_OPCODE_SUB
Numerical codes of operations, 29
KTC_OPCODE_SUBASSIGN
Numerical codes of operations, 29
KTC_OPCODE_THROW
Numerical codes of operations, 29
KTC_OPCODE_XORASSIGN
Numerical codes of operations, 29
KTC_POINTEROPERATOR_NONE
Numerical codes for identifying declarator ptr types, 40
KTC_POINTEROPERATOR_POINTER
Numerical codes for identifying declarator ptr types, 40
KTC_POINTEROPERATOR_REFERENCE
Numerical codes for identifying declarator ptr types, 40
KTC_POINTEROPERATOR_RVALUE
Numerical codes for identifying declarator ptr types, 40
ktc_position
ktcMainAPI.h, 261
ktc_position_copy
ktcMainAPI.h, 286
ktc_position_delete
ktcMainAPI.h, 287
ktc_position_getColumn
ktcMainAPI.h, 287
ktc_position_getFileName
ktcMainAPI.h, 287
ktc_position_getLine
ktcMainAPI.h, 287
ktc_position_new
ktcMainAPI.h, 287
ktc_position_setLine
ktcMainAPI.h, 288
ktc_position_t
ktcMainAPI.h, 262
ktc_proceed
ktcMainAPI.h, 288
ktc_registerRestoreContextHook
ktcMainAPI.h, 288
ktc_registerSaveContextHook
ktcMainAPI.h, 288
ktc_registerStartTraverseHook
ktcMainAPI.h, 289
ktc_registerStopTraverseHook
ktcMainAPI.h, 289
ktc_registerTreeHook
ktcMainAPI.h, 289
ktc_require
ktcMainAPI.h, 260
ktc_sema_checkBitField
ktcMainAPI.h, 289
ktc_sema_findAllByName
ktcMainAPI.h, 289
ktc_sema_findFirstByName
ktcMainAPI.h, 290
ktc_sema_forAllClassDeclarations
ktcMainAPI.h, 290
ktc_sema_forAllScopeDeclarations
ktcMainAPI.h, 291
ktc_sema_forAllSubtreeNodes
ktcMainAPI.h, 291
ktc_sema_functionOverloadsFunction
ktcMainAPI.h, 291
ktc_sema_getAllByName
ktcMainAPI.h, 292
ktc_sema_getArrayElementType
ktcMainAPI.h, 292
ktc_sema_getArraySize
ktcMainAPI.h, 292
ktc_sema_getAST
ktcMainAPI.h, 293
ktc_sema_getBaseInfo
ktcMainAPI.h, 293
ktc_sema_getBuiltinCode
ktcMainAPI.h, 293
ktc_sema_getClassTag
ktcMainAPI.h, 294
ktc_sema_getCVQualifiers
ktcMainAPI.h, 294
ktc_sema_getDefinedType
ktcMainAPI.h, 294
ktc_sema_getEnumerators
ktcMainAPI.h, 294
ktc_sema_getEnumUnderlyingTypeOrInt
ktcMainAPI.h, 295
ktc_sema_getFieldNumber
ktcMainAPI.h, 295
ktc_sema_getFieldType
ktcMainAPI.h, 295
ktc_sema_getFirstByName
ktcMainAPI.h, 296
ktc_sema_getFormalArgument
ktcMainAPI.h, 296
ktc_sema_getFunctionFromTemplate
ktcMainAPI.h, 296
ktc_sema_getFunctionPointerType
ktcMainAPI.h, 297
ktc_sema_getFunctionTemplateInstantiations
ktcMainAPI.h, 297
ktc_sema_getFunctionTemplateSpecializations
ktcMainAPI.h, 297
ktc_sema_getFunctionType
ktcMainAPI.h, 297
ktc_sema_getGlobalScope
ktcMainAPI.h, 297
ktc_sema_getIdentifier
ktcMainAPI.h, 298
ktc_sema_getIdentifierNo
ktcMainAPI.h, 298
ktc_sema_getInstantiatedClass
ktcMainAPI.h, 298

ktc_sema_getInstantiationOrigin
ktcMainAPI.h, 298
ktc_sema_getInstantiationParameters
ktcMainAPI.h, 298
ktc_sema_getIntegerValue
ktcMainAPI.h, 299
ktc_sema_getIntValueType
ktcMainAPI.h, 299
ktc_sema_getNumber
ktcMainAPI.h, 299
ktc_sema_getNumberOfArguments
ktcMainAPI.h, 300
ktc_sema_getNumberOfBaseInfo
ktcMainAPI.h, 300
ktc_sema_getObjectName
ktcMainAPI.h, 300
ktc_sema_getOverridenMethod
ktcMainAPI.h, 301
ktc_sema_getPointedType
ktcMainAPI.h, 301
ktc_sema_getPosition
ktcMainAPI.h, 301
ktc_sema_getPrimaryTemplate
ktcMainAPI.h, 301
ktc_sema_getQualifiedName
ktcMainAPI.h, 302
ktc_sema_getReferencedType
ktcMainAPI.h, 302
ktc_sema_getReturnType
ktcMainAPI.h, 302
ktc_sema_getScope
ktcMainAPI.h, 303
ktc_sema_getTypedefedName
ktcMainAPI.h, 303
ktc_sema_getTypeName
ktcMainAPI.h, 303
ktc_sema_getVariableInitializer
ktcMainAPI.h, 303
ktc_sema_getVariableType
ktcMainAPI.h, 304
ktc_sema_getVariableValue
ktcMainAPI.h, 304
ktc_sema_hasMethods
ktcMainAPI.h, 304
ktc_sema_haveSameFunctionType
ktcMainAPI.h, 304
ktc_sema_haveSameSignature
ktcMainAPI.h, 305
ktc_sema_isAnonymous
ktcMainAPI.h, 305
ktc_sema_isArray
ktcMainAPI.h, 305
ktc_sema_isBasePrivate
ktcMainAPI.h, 305
ktc_sema_isBaseProtected
ktcMainAPI.h, 306
ktc_sema_isBasePublic
ktcMainAPI.h, 306
ktc_sema_isBaseVirtual
ktcMainAPI.h, 306
ktc_sema_isBitfield
ktcMainAPI.h, 307
ktc_sema_isBuiltin
ktcMainAPI.h, 307
ktc_sema_isClass
ktcMainAPI.h, 307
ktc_sema_isConstMethod
ktcMainAPI.h, 307
ktc_sema_isConstructor
ktcMainAPI.h, 308
ktc_sema_isDestructor
ktcMainAPI.h, 308
ktc_sema_isEnum
ktcMainAPI.h, 308
ktc_sema_isEnumConstant
ktcMainAPI.h, 308
ktc_sema_isFriend
ktcMainAPI.h, 308
ktc_sema_isFuncDef
ktcMainAPI.h, 309
ktc_sema_isFunction
ktcMainAPI.h, 309
ktc_sema_isFunctionPointer
ktcMainAPI.h, 309
ktc_sema_isFunctionTemplate
ktcMainAPI.h, 309
ktc_sema_isFunctionTemplateSet
ktcMainAPI.h, 309
ktc_sema_isFunctionType
ktcMainAPI.h, 309
ktc_sema_isGlobal
ktcMainAPI.h, 310
ktc_sema_isImmutable
ktcMainAPI.h, 310
ktc_sema_isIncluded
ktcMainAPI.h, 310
ktc_sema_isInline
ktcMainAPI.h, 310
ktc_sema_isInstantiatedFunction
ktcMainAPI.h, 310
ktc_sema_isInstantiation
ktcMainAPI.h, 311
ktc_sema_isIntegerValue
ktcMainAPI.h, 311
ktc_sema_isLocal
ktcMainAPI.h, 311
ktc_sema_isNamespace
ktcMainAPI.h, 311
ktc_sema_isNamespaceAlias
ktcMainAPI.h, 311
ktc_sema_isNone
ktcMainAPI.h, 311
ktc_sema_isObjectValue
ktcMainAPI.h, 311
ktc_sema_isOperatorFunction
ktcMainAPI.h, 312

ktc_sema_isPOD
ktcMainAPI.h, 312
ktc_sema_isPointer
ktcMainAPI.h, 312
ktc_sema_isPrivate
ktcMainAPI.h, 312
ktc_sema_isProtected
ktcMainAPI.h, 312
ktc_sema_isPublic
ktcMainAPI.h, 312
ktc_sema_isPureVirtual
ktcMainAPI.h, 312
ktc_sema_isReference
ktcMainAPI.h, 313
ktc_sema_isSameClass
ktcMainAPI.h, 313
ktc_sema_isSameEnum
ktcMainAPI.h, 313
ktc_sema_isSameFunctions
ktcMainAPI.h, 313
ktc_sema_isSameScope
ktcMainAPI.h, 313
ktc_sema_isSameVariables
ktcMainAPI.h, 314
ktc_sema_isScope
ktcMainAPI.h, 314
ktc_sema_isSpecialization
ktcMainAPI.h, 314
ktc_sema_isStatic
ktcMainAPI.h, 314
ktc_sema_isTemplate
ktcMainAPI.h, 314
ktc_sema_isType
ktcMainAPI.h, 315
ktc_sema_isTypeParameter
ktcMainAPI.h, 315
ktc_sema_isUnion
ktcMainAPI.h, 315
ktc_sema_isUsing
ktcMainAPI.h, 315
ktc_sema_isUsingDeclaration
ktcMainAPI.h, 315
ktc_sema_isUsingDirective
ktcMainAPI.h, 315
ktc_sema_isVariable
ktcMainAPI.h, 315
ktc_sema_isVirtual
ktcMainAPI.h, 316
ktc_sema_RelatedClasses
ktcMainAPI.h, 316
ktc_sema_skipTypedefs
ktcMainAPI.h, 316
ktc_sema_stripCVQ
ktcMainAPI.h, 316
ktc_semanticInfo_t
ktcMainAPI.h, 262
ktc_skipBrackets
ktcMainAPI.h, 317
KTC_STORAGECLASS_AUTO
Numerical codes of declaration storage class specifiers, 30
KTC_STORAGECLASS_EXTERN
Numerical codes of declaration storage class specifiers, 30
KTC_STORAGECLASS_MUTABLE
Numerical codes of declaration storage class specifiers, 30
KTC_STORAGECLASS_NONE
Numerical codes of declaration storage class specifiers, 30
KTC_STORAGECLASS_REGISTER
Numerical codes of declaration storage class specifiers, 30
KTC_STORAGECLASS_STATIC
Numerical codes of declaration storage class specifiers, 31
KTC_STORAGECLASS_THREADLOCAL
Numerical codes of declaration storage class specifiers, 31
KTC_STORAGECLASS_TYPEDEF
Numerical codes of declaration storage class specifiers, 31
ktc_string_delete
ktcMainAPI.h, 317
ktc_string_get_cstring
ktcMainAPI.h, 317
ktc_string_new
ktcMainAPI.h, 318
ktc_string_t
ktcMainAPI.h, 262
KTC_TREE_EVENT_ON_ENTER
ktcMainAPI.h, 320
KTC_TREE_EVENT_ON_LEAVE
ktcMainAPI.h, 320
KTC_TREE_EVENT_ON_NEXT
ktcMainAPI.h, 320
ktc_tree_t
ktcMainAPI.h, 262
ktc_treeHook_t
ktcMainAPI.h, 262
ktc_treeType_getName
ktcMainAPI.h, 318
ktc_treeType_t
ktcMainAPI.h, 262
KTC_TYPESIGNEDNESS_DEFAULT
ktcMainAPI.h, 320
KTC_TYPESIGNEDNESS_NONE
ktcMainAPI.h, 320
KTC_TYPESIGNEDNESS_SIGNED
ktcMainAPI.h, 321
KTC_TYPESIGNEDNESS_UNSIGNED
ktcMainAPI.h, 321
ktc_unregisterTreeHook
ktcMainAPI.h, 318
ktcAPI.h, 254
ktcMainAPI.h, 255

KTC_API_VERSION_MAJOR, 260
KTC_API_VERSION_MINOR, 260
KTC_API_VERSION_PATCHLEVEL, 260
ktc_assembleStringConstant, 262
ktc_autofix_addSegment, 264
ktc_autofix_delete, 264
ktc_autofix_new, 264
ktc_autofix_t, 260
ktc_childId_t, 261
ktc_compareSubtrees, 264
ktc_constructorName, 318
KTC_CUSTOM_TYPES, 260
ktc_destructorName, 319
KTC_DIALECT_ARM, 319
KTC_DIALECT_GHS, 319
KTC_DIALECT_GNU, 319
KTC_DIALECT_MS, 319
KTC_DIALECT_STD, 319
KTC_DIALECT_SUN, 319
KTC_DIALECT_TI, 319
ktc_error_getConfigurationParameter, 265
ktc_error_isEnabled, 265
ktc_event_new, 265
ktc_event_setParameter, 266
ktc_eventHook_t, 261
ktc_forAllSubtreeNodes, 266
ktc_free, 266
ktc_getAssociatedScope, 266
ktc_getBuiltinType, 267
ktc_getBuiltinTypeSize, 267
ktc_getCallArgument, 268
ktc_getCalledFunction, 268
ktc_getCastSpecifier, 268
ktc_getClassTag, 269
ktc_getEndPosition, 269
ktc_getFrontendDialect, 269
ktc_getFrontendLanguage, 269
ktc_getFunctionSpecifier, 270
ktc_getIdentifier, 270
ktc_getIdentifierNo, 270
ktc_getIntegerValue, 270
ktc_getLanguageType, 271
ktc_getNameDeclarator, 271
ktc_getNoIdent, 271
ktc_getNumberOfCallArguments, 271
ktc_getOperation, 272
ktc_getOutputFileName, 272
ktc_getPointedType, 272
ktc_getPointerOperator, 272
ktc_getPointerSize, 273
ktc_getSemanticInfo, 273
ktc_getSizeofArgument, 273
ktc_getStartPosition, 273
ktc_getStorageClass, 274
ktc_getStringConstantValue, 274
ktc_getTokens, 274
ktc_getTypeQualifiers, 274
ktc_hasBuiltinWideChar, 275
ktc_is_NoToken, 275
ktc_is_Token, 275
ktc_isCallTo, 276
ktc_isCharLiteral, 276
ktc_isConstructor, 276
ktc_isDeclaration, 277
ktc_isDefinition, 277
ktc_isIncluded, 277
ktc_isMacroExpansion, 277
ktc_isMacroExpansion2, 278
ktc_isNameIncluded, 278
ktc_isNullMacro, 278
ktc_isNullPointerConstant, 278
ktc_isOperationOverloaded, 279
ktc_isStatic, 279
ktc_isTreeType, 279
ktc_isUTF16String, 281
ktc_isUTF32String, 281
ktc_isWideString, 281
KTC_LANGUAGE_C, 320
KTC_LANGUAGE_CXX, 320
ktc_languageType_t, 261
ktc_languageTypeEffectiveSignedness, 282
ktc_languageTypeIsBuiltin, 282
ktc_languageTypeIsPointer, 282
ktc_languageTypeSignedness, 282
ktc_languageTypeSize, 283
ktc_long_long_t, 261
ktc_message, 261
ktc_message_addAnchorAttribute, 283
ktc_message_addAttribute, 283
ktc_message_addEvent, 284
ktc_message_addEventEx, 284
ktc_message_addTraceBySemanticsInfo, 284
ktc_message_delete, 284
ktc_message_new, 284
ktc_message_render, 285
ktc_message_render_wi_autofix, 285
ktc_message_setFunction, 285
ktc_message_setPosition, 285
ktc_message_setRecommendationFactor, 285
ktc_message_t, 261
ktc_message_unsetFunction, 286
ktc_nodeStackGet, 286
ktc_nodeStackTop, 286
ktc_position, 261
ktc_position_copy, 286
ktc_position_delete, 287
ktc_position_getColumn, 287
ktc_position_getFileName, 287
ktc_position_getLine, 287
ktc_position_new, 287
ktc_position_setLine, 288
ktc_position_t, 262
ktc_proceed, 288
ktc_registerRestoreContextHook, 288
ktc_registerSaveContextHook, 288
ktc_registerStartTraverseHook, 289

ktc_registerStopTraverseHook, 289
ktc_registerTreeHook, 289
ktc_require, 260
ktc_sema_checkBitField, 289
ktc_sema_findAllByName, 289
ktc_sema_findFirstByName, 290
ktc_sema_forAllClassDeclarations, 290
ktc_sema_forAllScopeDeclarations, 291
ktc_sema_forAllSubtreeNodes, 291
ktc_sema_functionOverloadsFunction, 291
ktc_sema_getAllByName, 292
ktc_sema_getArrayElementType, 292
ktc_sema_getArraySize, 292
ktc_sema_getAST, 293
ktc_sema_getBaseInfo, 293
ktc_sema_getBuiltinCode, 293
ktc_sema_getClassTag, 294
ktc_sema_getCVQualifiers, 294
ktc_sema_getDefinedType, 294
ktc_sema_getEnumerators, 294
ktc_sema_getEnumUnderlyingTypeOrInt, 295
ktc_sema_getFieldNumber, 295
ktc_sema_getFieldType, 295
ktc_sema_getFirstByName, 296
ktc_sema_getFormalArgument, 296
ktc_sema_getFunctionFromTemplate, 296
ktc_sema_getFunctionPointerType, 297
ktc_sema_getFunctionTemplateInstantiations, 297
ktc_sema_getFunctionTemplateSpecializations, 297
ktc_sema_getFunctionType, 297
ktc_sema_getGlobalScope, 297
ktc_sema_getIdentifier, 298
ktc_sema_getIdentifierNo, 298
ktc_sema_getInstantiatedClass, 298
ktc_sema_getInstantiationOrigin, 298
ktc_sema_getInstantiationParameters, 298
ktc_sema_getIntegerValue, 299
ktc_sema_getIntValueType, 299
ktc_sema_getNumber, 299
ktc_sema_getNumberOfArguments, 300
ktc_sema_getNumberOfBaseInfo, 300
ktc_sema_getObjectName, 300
ktc_sema_getOverridenMethod, 301
ktc_sema_getPointedType, 301
ktc_sema_getPosition, 301
ktc_sema_getPrimaryTemplate, 301
ktc_sema_getQualifiedName, 302
ktc_sema_getReferencedType, 302
ktc_sema_getReturnType, 302
ktc_sema_getScope, 303
ktc_sema_getTypedefedName, 303
ktc_sema_getTypeName, 303
ktc_sema_getVariableInitializer, 303
ktc_sema_getVariableType, 304
ktc_sema_getVariableValue, 304
ktc_sema_hasMethods, 304
ktc_sema_haveSameFunctionType, 304
ktc_sema_haveSameSignature, 305
ktc_sema_isAnonymous, 305
ktc_sema_isArray, 305
ktc_sema_isBasePrivate, 305
ktc_sema_isBaseProtected, 306
ktc_sema_isBasePublic, 306
ktc_sema_isBaseVirtual, 306
ktc_sema_isBitfield, 307
ktc_sema_isBuiltin, 307
ktc_sema_isClass, 307
ktc_sema_isConstMethod, 307
ktc_sema_isConstructor, 308
ktc_sema_isDestructor, 308
ktc_sema_isEnum, 308
ktc_sema_isEnumConstant, 308
ktc_sema_isFriend, 308
ktc_sema_isFuncDef, 309
ktc_sema_isFunction, 309
ktc_sema_isFunctionPointer, 309
ktc_sema_isFunctionTemplate, 309
ktc_sema_isFunctionTemplateSet, 309
ktc_sema_isFunctionType, 309
ktc_sema_isGlobal, 310
ktc_sema_isImmutable, 310
ktc_sema_isIncluded, 310
ktc_sema_isInline, 310
ktc_sema_isInstantiatedFunction, 310
ktc_sema_isInstantiation, 311
ktc_sema_isIntegerValue, 311
ktc_sema_isLocal, 311
ktc_sema_isNamespace, 311
ktc_sema_isNamespaceAlias, 311
ktc_sema_isNone, 311
ktc_sema_isObjectValue, 311
ktc_sema_isOperatorFunction, 312
ktc_sema_isPOD, 312
ktc_sema_isPointer, 312
ktc_sema_isPrivate, 312
ktc_sema_isProtected, 312
ktc_sema_isPublic, 312
ktc_sema_isPureVirtual, 312
ktc_sema_isReference, 313
ktc_sema_isSameClass, 313
ktc_sema_isSameEnum, 313
ktc_sema_isSameFunctions, 313
ktc_sema_isSameScope, 313
ktc_sema_isSameVariables, 314
ktc_sema_isScope, 314
ktc_sema_isSpecialization, 314
ktc_sema_isStatic, 314
ktc_sema_isTemplate, 314
ktc_sema_isType, 315
ktc_sema_isTypeParameter, 315
ktc_sema_isUnion, 315
ktc_sema_isUsing, 315
ktc_sema_isUsingDeclaration, 315
ktc_sema_isUsingDirective, 315
ktc_sema_isVariable, 315

ktc_sema_isVirtual, 316
ktc_sema_RelatedClasses, 316
ktc_sema_skipTypedefs, 316
ktc_sema_stripCVQ, 316
ktc_semanticInfo_t, 262
ktc_skipBrackets, 317
ktc_string_delete, 317
ktc_string_get_cstring, 317
ktc_string_new, 318
ktc_string_t, 262
KTC_TREE_EVENT_ON_ENTER, 320
KTC_TREE_EVENT_ON_LEAVE, 320
KTC_TREE_EVENT_ON_NEXT, 320
ktc_tree_t, 262
ktc_treeHook_t, 262
ktc_treeType_getName, 318
ktc_treeType_t, 262
KTC_TYPESIGNEDNESS_DEFAULT, 320
KTC_TYPESIGNEDNESS_NONE, 320
KTC_TYPESIGNEDNESS_SIGNED, 321
KTC_TYPESIGNEDNESS_UNSIGNED, 321
ktc_unregisterTreeHook, 318
kw_array_delete
kwapi.h, 324
kw_array_get
kwapi.h, 324
kw_array_size
kwapi.h, 324
kw_array_t
kwapi.h, 323
kw_size_t
kwapi.h, 323
kwapi.h, 321
ktc_error_getConfigurationParameter, 324
kw_array_delete, 324
kw_array_get, 324
kw_array_size, 324
kw_array_t, 323
kw_size_t, 323
kwapi_apitypes_t, 323
kwapi_cfgparam_errorIsEnabled, 324
kwapi_cfgparam_getCheckerErrors, 325
kwapi_cfgparam_getConfigurationParameter, 325
kwapi_cfgparam_getListLength, 325
kwapi_cfgparam_getListNodeByIndex, 325
kwapi_cfgparam_getListNodeByName, 326
kwapi_cfgparam_getListNodeByRegexMatchingName, 326
kwapi_cfgparam_getName, 326
kwapi_cfgparam_getParameterValue, 326
kwapi_cfgparam_getParameterValueFromList, 327
kwapi_cfgparam_getRootParameterList, 327
kwapi_cfgparam_getType, 327
kwapi_cfgparam_isParameter, 327
kwapi_cfgparam_t, 323
KWAPI_CSHARP, 324
KWAPI_CXX, 324
KWAPI_DECLARE, 322
KWAPI_DECLARE_CPP, 322
KWAPI_DECLARE_DATA, 322
KWAPI_DECLARE_NONSTD, 322
KWAPI_JAVA, 324
kwapi_langtypes_t, 323
KWAPI_LINKER, 323
KWAPI_NOLANG, 324
KWAPI_NONE, 323
KWAPI_PATH, 323
KWAPI_PATTERN, 323
KWAPI_PREP, 323
KWAPI_TREE, 323
kwapi_apitypes_t
kwapi.h, 323
kwapi_cfgparam_errorIsEnabled
kwapi.h, 324
kwapi_cfgparam_getCheckerErrors
kwapi.h, 325
kwapi_cfgparam_getConfigurationParameter
kwapi.h, 325
kwapi_cfgparam_getListLength
kwapi.h, 325
kwapi_cfgparam_getListNodeByIndex
kwapi.h, 325
kwapi_cfgparam_getListNodeByName
kwapi.h, 326
kwapi_cfgparam_getListNodeByRegexMatchingName
kwapi.h, 326
kwapi_cfgparam_getName
kwapi.h, 326
kwapi_cfgparam_getParameterValue
kwapi.h, 326
kwapi_cfgparam_getParameterValueFromList
kwapi.h, 327
kwapi_cfgparam_getRootParameterList
kwapi.h, 327
kwapi_cfgparam_getType
kwapi.h, 327
kwapi_cfgparam_isParameter
kwapi.h, 327
kwapi_cfgparam_t
kwapi.h, 323
KWAPI_CSHARP
kwapi.h, 324
KWAPI_CXX
kwapi.h, 324
KWAPI_DECLARE
kwapi.h, 322
KWAPI_DECLARE_CPP
kwapi.h, 322
KWAPI_DECLARE_DATA
kwapi.h, 322
KWAPI_DECLARE_NONSTD
kwapi.h, 322
KWAPI_JAVA
kwapi.h, 324
kwapi_langtypes_t
kwapi.h, 323

KWAPI_LINKER
kwapi.h, 323
KWAPI_NOLANG
kwapi.h, 324
KWAPI_NONE
kwapi.h, 323
KWAPI_PATH
kwapi.h, 323
KWAPI_PATTERN
kwapi.h, 323
KWAPI_PREP
kwapi.h, 323
KWAPI_TREE
kwapi.h, 323

Numerical codes for identifying declarator ptr types, 40
ktc_get_child_index, 40
KTC_POINTEROPERATOR_NONE, 40
KTC_POINTEROPERATOR_POINTER, 40
KTC_POINTEROPERATOR_REFERENCE, 40
KTC_POINTEROPERATOR_RVALUE, 40
Numerical codes for identifying different C++ -style cast expressions, 39
KTC_CASTSPECIFIER_CONST, 39
KTC_CASTSPECIFIER_DYNAMIC, 39
KTC_CASTSPECIFIER_REINTERPRET, 39
KTC_CASTSPECIFIER_STATIC, 39
Numerical codes for identifying inline/virtual/explicit/friend member function specifiers, 38
KTC_FUNCSPECIFIER_EXPLICIT, 38
KTC_FUNCSPECIFIER_FRIEND, 38
KTC_FUNCSPECIFIER_INLINE, 38
KTC_FUNCSPECIFIER_NONE, 38
KTC_FUNCSPECIFIER_VIRTUAL, 38
Numerical codes of C/C++ built-in types, 34
KTC_BUILTINTYPE_BOOL, 34
KTC_BUILTINTYPE_CHAR, 34
KTC_BUILTINTYPE_DOUBLE, 34
KTC_BUILTINTYPE_FLOAT, 35
KTC_BUILTINTYPE_INT, 35
KTC_BUILTINTYPE_LONGDOUBLE, 35
KTC_BUILTINTYPE_LONGINT, 35
KTC_BUILTINTYPE_LONGLONGINT, 35
KTC_BUILTINTYPE_NONE, 35
KTC_BUILTINTYPE_NULLPTR_T, 35
KTC_BUILTINTYPE_SHORTINT, 35
KTC_BUILTINTYPE_SIGNEDCHAR, 36
KTC_BUILTINTYPE_SIGNEDINT, 36
KTC_BUILTINTYPE_SIGNEDLONGINT, 36
KTC_BUILTINTYPE_SIGNEDLONGLONGINT, 36
KTC_BUILTINTYPE_SIGNEDSHORTINT, 36
KTC_BUILTINTYPE_UNSIGNEDCHAR, 36
KTC_BUILTINTYPE_UNSIGNEDINT, 36
KTC_BUILTINTYPE_UNSIGNEDLONGINT, 36
KTC_BUILTINTYPE_UNSIGNEDLONGLONGINT, 37
KTC_BUILTINTYPE_UNSIGNEDSHORTINT, 37
KTC_BUILTINTYPE_VOID, 37
KTC_BUILTINTYPE_WCHAR_T, 37
Numerical codes of declaration storage class specifiers, 30
KTC_STORAGECLASS_AUTO, 30
KTC_STORAGECLASS_EXTERN, 30
KTC_STORAGECLASS_MUTABLE, 30
KTC_STORAGECLASS_NONE, 30
KTC_STORAGECLASS_REGISTER, 30
KTC_STORAGECLASS_STATIC, 31
KTC_STORAGECLASS_THREADLOCAL, 31
KTC_STORAGECLASS_TYPEDEF, 31
Numerical codes of declaration type qualifiers (no qualifier, 'const', 'volatile' or 'restrict'). Actual values are bit-or'ed superpositions of these flags. Use '==' check for KTC_CVQUALIFIER_NONE and '&' for two other values, 32
KTC_CVQUALIFIER_CONST, 32
KTC_CVQUALIFIER_NONE, 32
KTC_CVQUALIFIER_RESTRICT, 32
KTC_CVQUALIFIER_VOLATILE, 32
Numerical codes of operations, 22
KTC_OPCODE_ADD, 23
KTC_OPCODE_ADDASSIGN, 23
KTC_OPCODE_ADDRESS, 23
KTC_OPCODE_ANDASSIGN, 23
KTC_OPCODE_ASL, 23
KTC_OPCODE_ASLASSIGN, 23
KTC_OPCODE_ASR, 24
KTC_OPCODE_ASRASSIGN, 24
KTC_OPCODE_ASSIGN, 24
KTC_OPCODE_BITAND, 24
KTC_OPCODE_BITNOT, 24
KTC_OPCODE_BITOR, 24
KTC_OPCODE_BITXOR, 24
KTC_OPCODE_COMMA, 24
KTC_OPCODE_COND, 25
KTC_OPCODE_DEREF, 25
KTC_OPCODE_DEREFAST, 25
KTC_OPCODE_DIV, 25
KTC_OPCODE_DIVASSIGN, 25
KTC_OPCODE_DOTAST, 25
KTC_OPCODE_EQ, 25
KTC_OPCODE_FIELD, 25
KTC_OPCODE_FIELDREF, 26
KTC_OPCODE_GE, 26
KTC_OPCODE_GT, 26
KTC_OPCODE_LE, 26
KTC_OPCODE_LOGAND, 26
KTC_OPCODE_LOGNOT, 26
KTC_OPCODE_LOGOR, 26
KTC_OPCODE_LT, 26
KTC_OPCODE_MAX, 27
KTC_OPCODE_MIN, 27
KTC_OPCODE_MINUS, 27
KTC_OPCODE_MOD, 27
KTC_OPCODE_MODASSIGN, 27
KTC_OPCODE_MUL, 27
KTC_OPCODE_MULASSIGN, 27

KTC_OPCODE_NE, 27
KTC_OPCODE_NONE, 28
KTC_OPCODE_ORASSIGN, 28
KTC_OPCODE_PLUS, 28
KTC_OPCODE_POSTDEC, 28
KTC_OPCODE_POSTINC, 28
KTC_OPCODE_PREDEC, 28
KTC_OPCODE_PREINC, 28
KTC_OPCODE_ROUND_BRACKETS, 28
KTC_OPCODE_SIZEOF, 29
KTC_OPCODE_SQUARE_BRACKETS, 29
KTC_OPCODE_SUB, 29
KTC_OPCODE_SUBASSIGN, 29
KTC_OPCODE_THROW, 29
KTC_OPCODE_XORASSIGN, 29
Numerical codes to differentiate struct/class/union declarations, 33
KTC_CLASSTAG_CLASS, 33
KTC_CLASSTAG_NONE, 33
KTC_CLASSTAG_STRUCT, 33
KTC_CLASSTAG_UNION, 33

Obtaining configuration parameters for an error, 7

Setting and clearing handlers for events during tree traversal, 10

tid_AccessSpecification
gen-ktcAPI.h, 223
tid_AliasDecl
gen-ktcAPI.h, 223
tid_AlignAsExpr
gen-ktcAPI.h, 223
tid_AlignAsType
gen-ktcAPI.h, 223
tid_AlignOfExpr
gen-ktcAPI.h, 223
tid_Any
gen-ktcAPI.h, 224
tid_AnyAttribute
gen-ktcAPI.h, 224
tid_AnyCapture
gen-ktcAPI.h, 224
tid_AnyDecl
gen-ktcAPI.h, 224
tid_AnyDeclarator
gen-ktcAPI.h, 224
tid_AnyDesignator
gen-ktcAPI.h, 224
tid_AnyEnumerator
gen-ktcAPI.h, 224
tid_AnyExpr
gen-ktcAPI.h, 224
tid_AnyFuncBody
gen-ktcAPI.h, 225
tid_AnyInitializer
gen-ktcAPI.h, 225
tid_AnyLabel
gen-ktcAPI.h, 225
tid_AnyMemberDecl
gen-ktcAPI.h, 225
tid_AnyName
gen-ktcAPI.h, 225
tid_AnyNameQualifier
gen-ktcAPI.h, 225
tid_AnyNames
gen-ktcAPI.h, 225
tid_AnyNameSpec
gen-ktcAPI.h, 225
tid_AnyNonPtrDeclarator
gen-ktcAPI.h, 226
tid_AnyParamName
gen-ktcAPI.h, 226
tid_AnyPropertyFunc
gen-ktcAPI.h, 226
tid_AnyPseudoDtor
gen-ktcAPI.h, 226
tid_AnyStmt
gen-ktcAPI.h, 226
tid_AnyTemplateArg
gen-ktcAPI.h, 226
tid_AnyTypeName
gen-ktcAPI.h, 226
tid_AnyTypeOf
gen-ktcAPI.h, 226
tid_AnyTypeParam
gen-ktcAPI.h, 227
tid_AnyUsing
gen-ktcAPI.h, 227
tid_ArrayDeclarator
gen-ktcAPI.h, 227
tid_AsmDef
gen-ktcAPI.h, 227
tid_AsmStmt
gen-ktcAPI.h, 227
tid_Attribute
gen-ktcAPI.h, 227
tid_AttributedDeclarator
gen-ktcAPI.h, 227
tid_AttributeDeclSpec
gen-ktcAPI.h, 227
tid_Attributes
gen-ktcAPI.h, 228
tid_AttributeSpec
gen-ktcAPI.h, 228
tid_AttributeSpecs
gen-ktcAPI.h, 228
tid_AttributeWithArgs
gen-ktcAPI.h, 228
tid_AutoType
gen-ktcAPI.h, 228
tid_BaseSpec
gen-ktcAPI.h, 228
tid_BaseSpecs
gen-ktcAPI.h, 228
tid_BinaryExpr
gen-ktcAPI.h, 228

tid_BitFieldDeclarator
gen-ktcAPI.h, 229
tid_BoolLiteralExpr
gen-ktcAPI.h, 229
tid_BreakStmt
gen-ktcAPI.h, 229
tid_BuiltinType
gen-ktcAPI.h, 229
tid_CallExpr
gen-ktcAPI.h, 229
tid_Capture
gen-ktcAPI.h, 229
tid_CaptureDefault
gen-ktcAPI.h, 229
tid_CaseLabel
gen-ktcAPI.h, 229
tid_CaseRangeLabel
gen-ktcAPI.h, 230
tid_CastExpr
gen-ktcAPI.h, 230
tid_ClassType
gen-ktcAPI.h, 230
tid_CompoundStmt
gen-ktcAPI.h, 230
tid_ConditionalExpr
gen-ktcAPI.h, 230
tid_ConstExpr
gen-ktcAPI.h, 230
tid_ContinueStmt
gen-ktcAPI.h, 230
tid_ConvFunc
gen-ktcAPI.h, 230
tid_CopyInitializer
gen-ktcAPI.h, 231
tid_CtorInitializer
gen-ktcAPI.h, 231
tid_CVQualifier
gen-ktcAPI.h, 231
tid_Decl
gen-ktcAPI.h, 231
tid_DeclEllipsis
gen-ktcAPI.h, 231
tid_DeclOrStmt
gen-ktcAPI.h, 231
tid_DeclOrStmts
gen-ktcAPI.h, 231
tid_DeclSpec
gen-ktcAPI.h, 231
tid_DeclSpecs
gen-ktcAPI.h, 232
tid_DefaultException
gen-ktcAPI.h, 232
tid_DefaultLabel
gen-ktcAPI.h, 232
tid_DeleteExpr
gen-ktcAPI.h, 232
tid_DenyThrowSpec
gen-ktcAPI.h, 232
tid_Designators
gen-ktcAPI.h, 232
tid_DirectInitializer
gen-ktcAPI.h, 232
tid_DoDeclStmt
gen-ktcAPI.h, 232
tid_DoStmt
gen-ktcAPI.h, 233
tid_Dtor
gen-ktcAPI.h, 233
tid_Enumerator
gen-ktcAPI.h, 233
tid_Enumerators
gen-ktcAPI.h, 233
tid_EnumType
gen-ktcAPI.h, 233
tid_ExceptHandler
gen-ktcAPI.h, 233
tid_Exception
gen-ktcAPI.h, 233
tid_ExceptionSpec
gen-ktcAPI.h, 233
tid_ExplicitInstantiation
gen-ktcAPI.h, 234
tid_ExprArg
gen-ktcAPI.h, 234
tid_Exprs
gen-ktcAPI.h, 234
tid_ExprStmt
gen-ktcAPI.h, 234
tid_ExprTypeIdExpr
gen-ktcAPI.h, 234
tid_FieldDesignator
gen-ktcAPI.h, 234
tid_FinallyHandler
gen-ktcAPI.h, 234
tid_ForEachStmt
gen-ktcAPI.h, 234
tid_ForRangeStmt
gen-ktcAPI.h, 235
tid_ForStmt
gen-ktcAPI.h, 235
tid_FuncBody
gen-ktcAPI.h, 235
tid_FuncDeclarator
gen-ktcAPI.h, 235
tid_FuncDef
gen-ktcAPI.h, 235
tid_FuncSpec
gen-ktcAPI.h, 235
tid_FuncTryBlock
gen-ktcAPI.h, 235
tid_GenericAttribute
gen-ktcAPI.h, 235
tid_GlobalScope
gen-ktcAPI.h, 236
tid_GotoStmt
gen-ktcAPI.h, 236

tid_Handler
gen-ktcAPI.h, 236
tid_Handlers
gen-ktcAPI.h, 236
tid_IdExpr
gen-ktcAPI.h, 236
tid_IfDeclStmt
gen-ktcAPI.h, 236
tid_IfStmt
gen-ktcAPI.h, 236
tid_IndexDesignator
gen-ktcAPI.h, 236
tid_IndexExpr
gen-ktcAPI.h, 237
tid_InitClause
gen-ktcAPI.h, 237
tid_InitializedDeclarator
gen-ktcAPI.h, 237
tid_InitializerExpr
gen-ktcAPI.h, 237
tid_Initializers
gen-ktcAPI.h, 237
tid_KRFuncDeclarator
gen-ktcAPI.h, 237
tid_Label
gen-ktcAPI.h, 237
tid_LabeledStmt
gen-ktcAPI.h, 237
tid_LambdaDeclarator
gen-ktcAPI.h, 238
tid_LambdaExpr
gen-ktcAPI.h, 238
tid_LambdaIntroducer
gen-ktcAPI.h, 238
tid_LeaveStmt
gen-ktcAPI.h, 238
tid_LinkageSpec
gen-ktcAPI.h, 238
tid_LiteralExpr
gen-ktcAPI.h, 238
tid_MaybeCtorInitializer
gen-ktcAPI.h, 238
tid_MaybeDeclarator
gen-ktcAPI.h, 238
tid_MaybeException
gen-ktcAPI.h, 239
tid_MaybeExceptionSpec
gen-ktcAPI.h, 239
tid_MaybeLambdaDeclarator
gen-ktcAPI.h, 239
tid_MaybeNewInitializer
gen-ktcAPI.h, 239
tid_MaybeTypeId
gen-ktcAPI.h, 239
tid_MemberDecl
gen-ktcAPI.h, 239
tid_MemberDecls
gen-ktcAPI.h, 239
tid_MemberDesignator
gen-ktcAPI.h, 239
tid_MemberExpr
gen-ktcAPI.h, 240
tid_MemberFunc
gen-ktcAPI.h, 240
tid_MemberInitializer
gen-ktcAPI.h, 240
tid_MemberInitializers
gen-ktcAPI.h, 240
tid_MemberTemplate
gen-ktcAPI.h, 240
tid_MemberUsingDecl
gen-ktcAPI.h, 240
tid_Name
gen-ktcAPI.h, 240
tid_NameDeclarator
gen-ktcAPI.h, 240
tid_NamespaceAlias
gen-ktcAPI.h, 241
tid_NamespaceDecl
gen-ktcAPI.h, 241
tid_NameSpec
gen-ktcAPI.h, 241
tid_NewExpr
gen-ktcAPI.h, 241
tid_NewInitializer
gen-ktcAPI.h, 241
tid_NoAttribute
gen-ktcAPI.h, 241
tid_NoAttributeSpec
gen-ktcAPI.h, 241
tid_NoBaseSpec
gen-ktcAPI.h, 241
tid_NoCapture
gen-ktcAPI.h, 242
tid_NoCtorInitializer
gen-ktcAPI.h, 242
tid_Node
gen-ktcAPI.h, 242
tid_NoDeclarator
gen-ktcAPI.h, 242
tid_NoDeclOrStmt
gen-ktcAPI.h, 242
tid_NoDeclSpec
gen-ktcAPI.h, 242
tid_NoDesignator
gen-ktcAPI.h, 242
tid_NoEnumerator
gen-ktcAPI.h, 242
tid_NoException
gen-ktcAPI.h, 243
tid_NoExceptionSpec
gen-ktcAPI.h, 243
tid_NoExpr
gen-ktcAPI.h, 243
tid_NoHandler
gen-ktcAPI.h, 243

tid_NoInitializer
gen-ktcAPI.h, 243
tid_NoLambdaDeclarator
gen-ktcAPI.h, 243
tid_NoMemberDecl
gen-ktcAPI.h, 243
tid_NoMemberInitializer
gen-ktcAPI.h, 243
tid_NoName
gen-ktcAPI.h, 244
tid_NoNameQualifier
gen-ktcAPI.h, 244
tid_NoNewInitializer
gen-ktcAPI.h, 244
tid_NoParamName
gen-ktcAPI.h, 244
tid_NoPropertyFunc
gen-ktcAPI.h, 244
tid_NoTemplateArg
gen-ktcAPI.h, 244
tid_NoTemplateParam
gen-ktcAPI.h, 244
tid_NoTypeId
gen-ktcAPI.h, 244
tid_NullptrLiteralExpr
gen-ktcAPI.h, 245
tid_OpFunc
gen-ktcAPI.h, 245
tid_Param
gen-ktcAPI.h, 245
tid_ParamName
gen-ktcAPI.h, 245
tid_ParamNames
gen-ktcAPI.h, 245
tid_ParensDeclarator
gen-ktcAPI.h, 245
tid_ParensExpr
gen-ktcAPI.h, 245
tid_PromisedFuncBody
gen-ktcAPI.h, 245
tid_PromisedMemberDecl
gen-ktcAPI.h, 246
tid_PropertyAttribute
gen-ktcAPI.h, 246
tid_PropertyFuncs
gen-ktcAPI.h, 246
tid_PropertyGetFunc
gen-ktcAPI.h, 246
tid_PropertyPutFunc
gen-ktcAPI.h, 246
tid_PseudoDtor
gen-ktcAPI.h, 246
tid_PtrDeclarator
gen-ktcAPI.h, 246
tid_QualifiedName
gen-ktcAPI.h, 246
tid_QualifiedPseudoDtor
gen-ktcAPI.h, 247
tid_RangeDesignator
gen-ktcAPI.h, 247
tid_ReservedTypeSpec
gen-ktcAPI.h, 247
tid_ReturnStmt
gen-ktcAPI.h, 247
tid_SizeOfExpr
gen-ktcAPI.h, 247
tid_SpecialCastExpr
gen-ktcAPI.h, 247
tid_StaticAssertDecl
gen-ktcAPI.h, 247
tid_StmtExpr
gen-ktcAPI.h, 247
tid_StorageClass
gen-ktcAPI.h, 248
tid_StringLiteralExpr
gen-ktcAPI.h, 248
tid_SuffixFunc
gen-ktcAPI.h, 248
tid_SuperScope
gen-ktcAPI.h, 248
tid_SwitchDeclStmt
gen-ktcAPI.h, 248
tid_SwitchStmt
gen-ktcAPI.h, 248
tid_TemplateArgs
gen-ktcAPI.h, 248
tid_TemplateDecl
gen-ktcAPI.h, 248
tid_TemplateName
gen-ktcAPI.h, 249
tid_TemplateParam
gen-ktcAPI.h, 249
tid_TemplateParams
gen-ktcAPI.h, 249
tid_TemplateSpec
gen-ktcAPI.h, 249
tid_TemplateTypeArg
gen-ktcAPI.h, 249
tid_TemplateTypeParam
gen-ktcAPI.h, 249
tid_ThisExpr
gen-ktcAPI.h, 249
tid_ThrowExpr
gen-ktcAPI.h, 249
tid_TranslationUnit
gen-ktcAPI.h, 250
tid_TruncatedInitClause
gen-ktcAPI.h, 250
tid_TryExceptStmt
gen-ktcAPI.h, 250
tid_TryFinallyStmt
gen-ktcAPI.h, 250
tid_TryStmt
gen-ktcAPI.h, 250
tid_TypeAdjective
gen-ktcAPI.h, 250

tid_TypeArg
gen-ktcAPI.h, 250
tid_TypeConvExpr
gen-ktcAPI.h, 250
tid_TypeId
gen-ktcAPI.h, 251
tid_TypeName
gen-ktcAPI.h, 251
tid_TypeOfExpr
gen-ktcAPI.h, 251
tid_TypeOfSpec
gen-ktcAPI.h, 251
tid_TypeOfType
gen-ktcAPI.h, 251
tid_TypeParam
gen-ktcAPI.h, 251
tid_TypeTypeIdExpr
gen-ktcAPI.h, 251
tid_UnaryExpr
gen-ktcAPI.h, 251
tid_UnparsedDecl
gen-ktcAPI.h, 252
tid_UnparsedDeclarator
gen-ktcAPI.h, 252
tid_UnparsedDeclSpec
gen-ktcAPI.h, 252
tid_UnparsedEnumerator
gen-ktcAPI.h, 252
tid_UnparsedException
gen-ktcAPI.h, 252
tid_UnparsedExpr
gen-ktcAPI.h, 252
tid_UnparsedInitializer
gen-ktcAPI.h, 252
tid_UnparsedLabel
gen-ktcAPI.h, 252
tid_UnparsedMemberDecl
gen-ktcAPI.h, 253
tid_UnparsedName
gen-ktcAPI.h, 253
tid_UnparsedNameQualifier
gen-ktcAPI.h, 253
tid_UnparsedParamName
gen-ktcAPI.h, 253
tid_UnparsedPropertyFunc
gen-ktcAPI.h, 253
tid_UnparsedStmt
gen-ktcAPI.h, 253
tid_UnqualifiedName
gen-ktcAPI.h, 253
tid_UserLiteralExpr
gen-ktcAPI.h, 253
tid_UserStringLiteralExpr
gen-ktcAPI.h, 254
tid_UsingDecl
gen-ktcAPI.h, 254
tid_UsingDirective
gen-ktcAPI.h, 254
tid_WhileDeclStmt
gen-ktcAPI.h, 254
tid_WhileStmt
gen-ktcAPI.h, 254
Tree type checking predicates, 20
Tree type identifiers, 19

Utility functions for tree access, 14

Working with tree positions, 15
Working with warning and error messages, 18