

Global Index

\$ modification : See Section “Modifiers” in *syntax*
& marker for actions : See Section “Actions” in *syntax*
+= : See Section “Macros” in *fw*
-R : See Section “Upgrading” in *faq*
-r : See Section “CacheSize” in *faq*
.FINL.phi files : See Section “Implementing” in *comptrees*
.HEAD.phi files : See Section “Implementing” in *comptrees*
.INIT.phi files : See Section “Implementing” in *comptrees*
.c files : See Section “Implementing” in *comptrees*
.clp : See Section “Command line format” in *clp*
.clp : See Section “Specification” in *clp*
.con files : See Section “Tree Construction” in *comptrees*
.finl files : See Section “Implementing” in *comptrees*
.finl files : See Section “Refactor” in *news4.8*
.fw : See *fw*
.fwi : See Section “Include Files” in *fw*
.fwi : See *fw*
.h files : See Section “Implementing” in *comptrees*
.head files : See Section “Implementing” in *comptrees*
.init files : See Section “Implementing” in *comptrees*
.init files : See Section “Refactor” in *news4.8*
.lido files : See Section “LIGA Files” in *comptrees*
.pdl : See Section “Properties” in *deftbl*
.specs files : See Section “Implementing” in *comptrees*
.sym files : See Section “Tree Construction” in *comptrees*
1 : See Section “Expert” in *info*
2 : See Section “Expert” in *info*
3 : See Section “Expert” in *info*
4 : See Section “Expert” in *info*
5 : See Section “Expert” in *info*
== : See Section “Macros” in *fw*
? : See Section “Help-C-I” in *info*
@ modification : See Section “Modifiers” in *syntax*
@! : See Section “Comments” in *fw*
@+ : See Section “Inserting EOL” in *fw*
@- : See Section “Suppressing End of Line” in *fw*
@1... : See Section “Parameter Lists” in *fw*
@M : See Section “Macros” in *fw*
@Z : See Section “Macros” in *fw*
@i : See Section “Include Files” in *fw*
! : See Section “Session” in *novice*
! : See Section “Odinfile” in *ui*
% : See Section “Session” in *novice*
% : See Section “Odinfile” in *ui*
* : See Section “Star” in *lex*

+arg and mongdb : See Section “Monitoring products” in *news4.3*
 +lang : See Section “DeriveTree” in *idem*
 +nomain : See Section “nomain” in *news4.4*
 +parser : See Section “nomain” in *news4.4*
 +patterns : See Section “Basis” in *idem*
 +prefix : See Section “Basis” in *idem*
 + : See Section “Plus” in *lex*
 -R command line parameter : See Section “Cache” in *novice*
 -c command line argument : See Section “Cache” in *novice*
 -g flag : See Section “debug” in *pp*
 -p parameter : See Section “EliCmd” in *news4.6*
 -r command line parameter : See Section “Cache” in *novice*
 -s : See Section “Implement” in *sysadmin*
 - : See Section “Classes” in *lex*
 .odinrc : See Section “Odinrc” in *faq*
 . : See Section “Classes” in *lex*
 == : See Section “Session” in *novice*
 == : See Section “Odinfile” in *ui*
 ? : See Section “Query” in *lex*
 { } : See Section “Count” in *lex*
 ADA_COMMENT : See Section “Available Descriptions” in *lex*
 ADA_COMMENT : See Section “Canned Descriptions” in *lex*
 ADA_COMMENT : See Section “Definitions of Descriptions” in *lex*
 ADA_IDENTIFIER : See Section “Available Descriptions” in *lex*
 ADA_IDENTIFIER : See Section “Definitions of Descriptions” in *lex*
 ADD : See Section “Predefined Entities” in *lidoref*
 AND : See Section “Predefined Entities” in *lidoref*
 APPLY : See Section “Predefined Entities” in *lidoref*
 ARGTOONE : See Section “Predefined Entities” in *lidoref*
 ATTR : See Section “Types and Classes of Attributes” in *lidoref*
 AWK_COMMENT : See Section “Available Descriptions” in *lex*
 AWK_COMMENT : See Section “Definitions of Descriptions” in *lex*
 AlgScopeProp : See Section “ScopeProp” in *news4.4*
 AlgScope module : See Section “Instantiation” in *novice*
 ArgError : See Section “ChkExpr” in *type*
 ArgErr : See Section “ChkExpr” in *type*
 ArgMsg : See Section “ChkExpr” in *type*
 AsIs, Ptg-Pattern : See Section “Frequently Used Patterns” in *output*
 BITAND : See Section “Predefined Entities” in *lidoref*
 BITOR : See Section “Predefined Entities” in *lidoref*
 BITXOR : See Section “Predefined Entities” in *lidoref*
 BOTTOMUP : See Section “Bottomup” in *comptrees*
 BOTTOMUP : See Section “Bottom-up Evaluation” in *news4.1*
 BOTTOM_UP : See Section “bottom-up evaluation” in *news4.0*
 BOTTOM_UP : See Section “Bottom-up Evaluation” in *news4.1*
 BP_BeginBlockI : See Section “treeseq” in *idem*
 BP_BreakLine : See Section “treeseq” in *idem*

BP_EndBlockI : See Section “treesep” in *idem*
 BP_OutFile : See Section “Multiple” in *idem*
 BP_Out : See Section “treesep” in *idem*
 Bind : See Section “Name” in *news4.3*
 BuScopeProp : See Section “ScopeProp” in *news4.4*
 BuildHosts : See Section “Variables” in *ui*
 CChar, Ptg-Pattern : See Section “Frequently Used Patterns” in *output*
 CHAINSTART : See Section “CHAIN” in *lidoref*
 CHAINSTART : See Section “separated CHAINSTART” in *news4.0*
 CHAINSTART : See Section “CHAIN in Empty Rules” in *news4.1*
 CHAIN : See Section “CHAIN” in *lidoref*
 CHAIN : See Section “Remote Attribute Access” in *lidoref*
 CLASS SYMBOL : See Section “class and tree symbols” in *news4.0*
 CLASS : See Section “Symbol Specifications” in *lidoref*
 CLASS symbols : See Section “CONSTITUENT(S)” in *lidoref*
 CLASS symbols : See Section “Inheritance of Computations” in *lidoref*
 CLASS symbols : See Section “Remote Attribute Access” in *lidoref*
 CLASS symbols : See Section “Symbol Specifications” in *lidoref*
 CLASS, OIL : See Section “OilClass” in *type*
 COERCION, OIL : See Section “Coercibility” in *type*
 COERCION, OIL : See Section “OilClass” in *type*
 COL : See Section “Predefined Entities” in *lidoref*
 COL : See Section “LINE COL COORDREF” in *news4.0*
 COL : See Section “Start” in *syntax*
 COMPUTE : See Section “Lido Tokens” in *news4.0*
 CONDITION : See Section “Lido Tokens” in *news4.0*
 CONSTITUENT: See [\[\]](#), page [\(lidoref\)CONSTITUENT\(S\),,,S,S](#)

 CONSTITUENT: See [\[\]](#), page [\(lidoref\)Remote Attribute Access,.,S,S](#)
 CONSTITUENTS : See Section “CONSTITUENT(S)” in *lidoref*
 CONSTITUENTS : See Section “Terminals” in *lidoref*
 CONSTITUENTS : See Section “CONSTITUENTS Restrictions Removed” in *news4.1*
 CONSTITUENTS : See Section “Output” in *novice*
 CONSTITUENT : See Section “CONSTITUENT(S)” in *lidoref*
 CONSTITUENT : See Section “Terminals” in *lidoref*
 COORDREF : See Section “Predefined Entities” in *lidoref*
 COORDREF : See Section “LINE COL COORDREF” in *news4.0*
 COORDREF : See Section “Start” in *syntax*
 CPP : See Section “treelng” in *idem*
 CPtgOutchar : See Section “Useful Embedded Functions” in *output*
 CPtgOutstr : See Section “Useful Embedded Functions” in *output*
 CScopeProp : See Section “ScopeProp” in *news4.4*
 CString, Ptg-Pattern : See Section “Frequently Used Patterns” in *output*
 C_CHAR_CONSTANT : See Section “Available Descriptions” in *lex*
 C_CHAR_CONSTANT : See Section “Definitions of Descriptions” in *lex*
 C_COMMENT : See Section “Available Descriptions” in *lex*

C_COMMENT : See Section “Definitions of Descriptions” in *lex*
C_COMMENT : See Section “GLA specification” in *novice*
C_FLOAT : See Section “Available Descriptions” in *lex*
C_FLOAT : See Section “Definitions of Descriptions” in *lex*
C_IDENTIFIER : See Section “Available Descriptions” in *lex*
C_IDENTIFIER : See Section “Canned Descriptions” in *lex*
C_IDENTIFIER : See Section “Definitions of Descriptions” in *lex*
C_INTEGER : See Section “Available Descriptions” in *lex*
C_INTEGER : See Section “Definitions of Descriptions” in *lex*
C_INT_DENOTATION : See Section “Available Descriptions” in *lex*
C_INT_DENOTATION : See Section “Definitions of Descriptions” in *lex*
C_STRING_LIT : See Section “C_STRING_LIT” in *faq*
C_STRING_LIT : See Section “Available Descriptions” in *lex*
C_STRING_LIT : See Section “Canned Descriptions” in *lex*
C_STRING_LIT : See Section “Definitions of Descriptions” in *lex*
C_STRING_LIT : See Section “Reproducing String Literals” in *output*
CommaSeq, Ptg-Pattern. : See Section “Frequently Used Patterns” in *output*
Counter : See Section “Tech” in *news4.3*
Ctext : See Section “Available scanners” in *lex*
DAPTO_ARG and non-standard types : See Section “Non-standard types” in *mon*
DAPTO_RESULT_INTVAL : See Section “Operations” in *mon*
DAPTO_RESULT_INT : See Section “Operations” in *mon*
DAPTO_RESULT_LONGVAL : See Section “Operations” in *mon*
DAPTO_RESULT_LONG : See Section “Operations” in *mon*
DAPTO_RESULT_PTRVAL : See Section “Operations” in *mon*
DAPTO_RESULT_PTR : See Section “Operations” in *mon*
DAPTO_RESULT_STR : See Section “Operations” in *mon*
DAPTO_RESULT and non-standard types : See Section “Non-standard types” in *mon*
DEPENDS_ON : See Section “depends on” in *news4.0*
DIV : See Section “Predefined Entities” in *lidoref*
Dir : See Section “Variables” in *ui*
EDITOR : See Section “Odinrc” in *faq*
EDITOR : See Section “Session” in *novice*
ELI_DEBUGGER. : See Section “Debugging” in *pp*
EOF : See Section “At” in *syntax*
EQ : See Section “Predefined Entities” in *lidoref*
ERROR : See Section “Instantiation” in *novice*
ERROR : See Section “Output” in *novice*
ElemToBitSet : See Section “Adt” in *news4.3*
Entity : See Section “Instantiation” in *novice*
Eol, Ptg-Pattern. : See Section “Frequently Used Patterns” in *output*
ErrLevel : See Section “Variables” in *ui*
ErrorCount : See Section “Output” in *novice*
Errors, lexical : See Section “Illegal” in *lex*
ExpError : See Section “ChkExpr” in *type*
ExpErr : See Section “ChkExpr” in *type*
ExpMsg : See Section “ChkExpr” in *type*

Expression module : See Section “Expressions” in *type*
 Expression module : See Section “IdentifyOperator” in *type*
 GENTREE : See Section “Computed Subtrees” in *lidoref*
 GENTREE : See Section “Predefined Entities” in *lidoref*
 GE : See Section “Predefined Entities” in *lidoref*
 GT : See Section “Predefined Entities” in *lidoref*
 HEAD : See Section “CHAIN” in *lidoref*
 HEAD : See Section “CHAIN in Empty Rules” in *news4.1*
 HelpLevel : See Section “Variables” in *ui*
 History : See Section “Variables” in *ui*
 IDENTICAL : See Section “Predefined Entities” in *lidoref*
 IF : See Section “Predefined Entities” in *lidoref*
 IGNORE macro : See Section “Combine” in *idem*
 INCLUDING : See Section “INCLUDING” in *lidoref*
 INCLUDING : See Section “Remote Attribute Access” in *lidoref*
 INCLUDING : See Section “Terminals” in *news4.0*
 INDICATION, OIL : See Section “Indications” in *type*
 INDICATION, OIL : See Section “OilClass” in *type*
 INHERITS : See Section “Inheritance of Computations” in *lidoref*
 INHERITS : See Section “Instantiation” in *novice*
 INH : See Section “Attributes” in *lidoref*
 INH : See Section “CHAIN” in *lidoref*
 INH : See Section “Symbol Specifications” in *lidoref*
 INH : See Section “Types and Classes of Attributes” in *lidoref*
 IS : See Section “chain productions” in *news4.0*
 ITERATE : See Section “Iterations” in *lidoref*
 IdDefScope : See Section “Instantiation” in *novice*
 IdemOrigPtg attribute : See Section “Compute” in *idem*
 IdemPtg : See Section “idemsep” in *idem*
 IdemPtg : See Section “treedel” in *idem*
 IdemPtg : See Section “treesep” in *idem*
 IdemPtg, computation : See Section “Changing” in *idem*
 IdemPtg, computation : See Section “Compute” in *idem*
 IdemPtg, overriding : See Section “Compute” in *idem*
 IdemReproduce_X : See Section “idemdel” in *idem*
 IdemReproduce_X : See Section “treedel” in *idem*
 IdemReproduce : See Section “idemsep” in *idem*
 IdemReproduce : See Section “treesep” in *idem*
 Id, Ptg Pattern : See Section “Frequently Used Patterns” in *output*
 Java : See Section “treelng” in *idem*
 KeepGoing : See Section “Variables” in *ui*
 LE : See Section “Predefined Entities” in *lidoref*
 LIGAPragma : See Section “Pragmas” in *lidoref*
 LIGAPragma : See Section “bottom-up evaluation” in *news4.0*
 LINE : See Section “Predefined Entities” in *lidoref*
 LINE : See Section “LINE COL COORDREF” in *news4.0*
 LINE : See Section “Start” in *syntax*

LISTEDTO : See Section “type definitions” in *news4.0*
 LISTOF : See Section “Productions” in *lidoref*
 LISTOF : See Section “Terminals” in *lidoref*
 LISTOF : See Section “terminals in LISTOF productions” in *news4.0*
 LISTOF production : See Section “Tree” in *comptrees*
 LISTOF productions : See Section “idem” in *idem*
 LISTOF productions : See Section “Tree Construction Functions” in *lidoref*
 LISTOF productions, computations : See Section “idemdel” in *idem*
 LISTOF productions, computations : See Section “treedel” in *idem*
 LT : See Section “Predefined Entities” in *lidoref*
 LeafPtg : See Section “Output” in *novice*
 Lexical errors : See Section “Illegal” in *lex*
 LidoList : See Section “Adt” in *news4.3*
 LineNum : See Section “Coordinates” in *lex*
 List : See Section “Adt” in *news4.3*
 LogLevel : See Section “Session” in *novice*
 LogLevel : See Section “procex” in *novice*
 LogLevel : See Section “Derivation” in *pp*
 LogLevel : See Section “Variables” in *ui*
 LstError : See Section “ChkExpr” in *type*
 LstErr : See Section “ChkExpr” in *type*
 LstMsg : See Section “ChkExpr” in *type*
 MODULA2_CHARINT : See Section “Available Descriptions” in *lex*
 MODULA2_CHARINT : See Section “Definitions of Descriptions” in *lex*
 MODULA2_COMMENT : See Section “Available Descriptions” in *lex*
 MODULA2_COMMENT : See Section “Definitions of Descriptions” in *lex*
 MODULA2_INTEGER : See Section “Available Descriptions” in *lex*
 MODULA2_INTEGER : See Section “Definitions of Descriptions” in *lex*
 MODULA2_LITERALDQ : See Section “Available Descriptions” in *lex*
 MODULA2_LITERALDQ : See Section “Definitions of Descriptions” in *lex*
 MODULA2_LITERALSQ : See Section “Available Descriptions” in *lex*
 MODULA2_LITERALSQ : See Section “Definitions of Descriptions” in *lex*
 MODULA3_COMMENT : See Section “Available Descriptions” in *lex*
 MODULA3_COMMENT : See Section “Definitions of Descriptions” in *lex*
 MODULA_INTEGER : See Section “Definitions of Descriptions” in *lex*
 MOD : See Section “Predefined Entities” in *lidoref*
 MONITOR : See Section “Return” in *lex*
 MUL : See Section “Predefined Entities” in *lidoref*
 MaxBuilds : See Section “Variables” in *ui*
 NEG : See Section “Predefined Entities” in *lidoref*
 NEW_LINE : See Section “Available Descriptions” in *lex*
 NEW_LINE : See Section “Definitions of Descriptions” in *lex*
 NE : See Section “Predefined Entities” in *lidoref*
 NODEPTR : See Section “Computed Subtrees” in *lidoref*
 NODEPTR : See Section “Predefined Entities” in *lidoref*
 NODEPTR : See Section “Tree Construction Functions” in *lidoref*
 NORETURN : See Section “Building Processors” in *lex*

NORETURN : See Section “Generated Module” in *lex*
 NOT : See Section “Predefined Entities” in *lidoref*
 NULLNODEPTR : See Section “Predefined Entities” in *lidoref*
 NULLNODEPTR : See Section “Tree Construction Functions” in *lidoref*
 NoKey : See Section “Property storage” in *novice*
 NoKey : See Section “Types” in *type*
 Numb, Ptg-Pattern : See Section “Frequently Used Patterns” in *output*
 ODINVERIFYLEVEL : See Section “ChangingFiles” in *faq*
 ODIN_LOCALIPC : See Section “IPC” in *faq*
 ONE : See Section “Predefined Entities” in *lidoref*
 OPER, OIL : See Section “OilClass” in *type*
 OPER, OIL : See Section “Operators” in *type*
 ORDER : See Section “Predefined Entities” in *lidoref*
 OR : See Section “Predefined Entities” in *lidoref*
 ObjCnt : See Section “Prop” in *news4.3*
 OilNoOverload : See Section “OilNoOverload” in *news4.4*
 OprError : See Section “ChkExpr” in *type*
 OprErr : See Section “ChkExpr” in *type*
 OprMsg : See Section “ChkExpr” in *type*
 PASCAL_COMMENT : See Section “Available Descriptions” in *lex*
 PASCAL_COMMENT : See Section “Definitions of Descriptions” in *lex*
 PASCAL_IDENTIFIER : See Section “Available Descriptions” in *lex*
 PASCAL_IDENTIFIER : See Section “Definitions of Descriptions” in *lex*
 PASCAL_INTEGER : See Section “Available Descriptions” in *lex*
 PASCAL_INTEGER : See Section “Canned Descriptions” in *lex*
 PASCAL_INTEGER : See Section “Definitions of Descriptions” in *lex*
 PASCAL_REAL : See Section “Available Descriptions” in *lex*
 PASCAL_REAL : See Section “Definitions of Descriptions” in *lex*
 PASCAL_STRING : See Section “Available Descriptions” in *lex*
 PASCAL_STRING : See Section “Canned Descriptions” in *lex*
 PASCAL_STRING : See Section “Definitions of Descriptions” in *lex*
 PPtgOutstr : See Section “Useful Embedded Functions” in *output*
 PString, Ptg-Pattern : See Section “Frequently Used Patterns” in *output*
 PTGAsIs: See [\[\]](#), page [\[\]](#) (output)Frequently Used Patterns,,,,
 PTGCCChar: See [\[\]](#), page [\[\]](#) (output)Frequently Used Patterns,,,,
 PTGCString: See [\[\]](#), page [\[\]](#) (output)Frequently Used Patterns,,,,
 PTGCStringId: See Section “Frequently Used Patterns” in *output*
 PTGId: See [\[\]](#), page [\[\]](#) (output)Frequently Used Patterns,,,,
 PTGNULL : See Section “Text generation” in *novice*
 PTGName : See Section “Output” in *novice*
 PTGNode : See Section “Output” in *novice*
 PTGNumb: See [\[\]](#), page [\[\]](#) (output)Frequently Used Patterns,,,,

PTGNumb : See Section “Output” in *novice*
 PTGPString: See [\[\]](#), page [\(output\)Frequently Used Patterns](#),
 PTGPStringId : See Section “Frequently Used Patterns” in *output*
 Pragma : See Section “Pragmas” in *lidoref*
 PreDefBind : See Section “Name” in *news4.3*
 PreDefId : See Section “Name” in *news4.3*
 PreDefKeyBind : See Section “Name” in *news4.3*
 PreDefSymKeyBind : See Section “Name” in *news4.3*
 PreDefine : See Section “Name” in *news4.3*
 PtgOutId : See Section “Combine” in *idem*
 PtgOutId : See Section “Useful Embedded Functions” in *output*
 PtrList : See Section “Adt” in *news4.3*
 RETURN : See Section “Return” in *lex*
 ROOTCLASS : See Section “Predefined Entities” in *lidoref*
 ROOTCLASS : See Section “Symbol Specifications” in *lidoref*
 ROOTCLASS : See Section “Grammar Root Symbol” in *news4.1*
 RULENAME : See Section “Predefined Entities” in *lidoref*
 RULENAME : See Section “Right-Hand Side Access” in *news4.1*
 RULE : See Section “Rule Specifications” in *lidoref*
 RangeScope : See Section “Instantiation” in *novice*
 RangeUnique : See Section “Instantiation” in *novice*
 Reporting a lexical error : See Section “Illegal” in *lex*
 ResetScan : See Section “Generated Module” in *lex*
 RhsAttrs : See Section “Attributes” in *lidoref*
 RhsFct : See Section “Right-Hand Side Access” in *news4.1*
 RootScope : See Section “Instantiation” in *novice*
 RuleName : See Section “Rule Specifications” in *lidoref*
 SCANPTR : See Section “Reset” in *lex*
 SETCOORD : See Section “Position” in *lex*
 SETENDCOORD : See Section “Position” in *lex*
 SHIELD : See Section “CONSTITUENT(S)” in *lidoref*
 SPACES : See Section “Available Descriptions” in *lex*
 SPACES : See Section “Skipping” in *lex*
 STATIC : See Section “Lido Tokens” in *news4.0*
 SUB : See Section “Predefined Entities” in *lidoref*
 SYMBOL : See Section “Symbol Specifications” in *lidoref*
 SYMBOL : See Section “Instantiation” in *novice*
 SYNT : See Section “Attributes” in *lidoref*
 SYNT : See Section “CHAIN” in *lidoref*
 SYNT : See Section “Symbol Specifications” in *lidoref*
 SYNT : See Section “Types and Classes of Attributes” in *lidoref*
 ScopeProp : See Section “ScopeProp” in *news4.4*
 Sep_Out : See Section “Multiple” in *idem*
 Sep_Out : See Section “Use” in *idem*
 Sep_Out : See Section “idemsep” in *idem*
 Sep_Out : See Section “Separator” in *output*

Sep_Print : See Section “idemsep” in *idem*
 Separator : See Section “Combine” in *idem*
 Seq, Ptg-Pattern : See Section “Frequently Used Patterns” in *output*
 Size : See Section “Variables” in *ui*
 StartLine : See Section “Coordinates” in *lex*
 StartLine : See Section “Generated Module” in *lex*
 StructEquiv module : See Section “TypeEq” in *type*
 TAB : See Section “Available Descriptions” in *lex*
 TAIL : See Section “CHAIN” in *lidoref*
 TAIL : See Section “CHAIN in Empty Rules” in *news4.1*
 TERM : See Section “Productions” in *lidoref*
 TERM : See Section “Terminal Access” in *lidoref*
 TERM : See Section “Terminals” in *lidoref*
 TERM : See Section “Terminals” in *news4.0*
 TERM : See Section “Instantiation” in *novice*
 THIS : See Section “Attributes” in *lidoref*
 THIS : See Section “CHAIN” in *lidoref*
 THIS : See Section “Symbol Specifications” in *lidoref*
 TRANSFER : See Section “TRANSFER” in *news4.0*
 TREE SYMBOL : See Section “class and tree symbols” in *news4.0*
 TREE_COMPLETE : See Section “Bottom-up Evaluation” in *news4.1*
 TREE : See Section “Symbol Specifications” in *lidoref*
 TREE symbols : See Section “CONSTITUENT(S)” in *lidoref*
 TREE symbols : See Section “Symbol Specifications” in *lidoref*
 TYPE : See Section “type definitions” in *news4.0*
 TermFct : See Section “Right-Hand Side Access” in *news4.1*
 TokenEnd : See Section “Generated Module” in *lex*
 TokenStart : See Section “Generated Module” in *lex*
 Typing module : See Section “BasicType” in *type*
 UNTIL : See Section “Iterations” in *lidoref*
 Unique module : See Section “Instantiation” in *novice*
 VOIDEN : See Section “Predefined Entities” in *lidoref*
 VOID : See Section “Attribute Computations and Plain Computations” in *lidoref*
 VOID : See Section “Iterations” in *lidoref*
 VOID : See Section “Predefined Entities” in *lidoref*
 VOID : See Section “Types and Classes of Attributes” in *lidoref*
 VOID context : See Section “Attribute Computations and Plain Computations” in *lidoref*
 VOID context : See Section “CONSTITUENT(S)” in *lidoref*
 VOID context : See Section “Dependent Expressions” in *lidoref*
 VOID context : See Section “INCLUDING” in *lidoref*
 VOID context : See Section “Simple Expressions” in *lidoref*
 VOID context : See Section “Types and Classes of Attributes” in *lidoref*
 VerifyLevel : See Section “ChangingFiles” in *faq*
 VerifyLevel : See Section “Variables” in *ui*
 WITH : See Section “CONSTITUENT(S)” in *lidoref*
 WRAPUPMONITOR : See Section “Continue” in *lex*
 WRAPUP : See Section “Continue” in *lex*

WarnLevel : See Section “Variables” in *ui*
 XML : See Section “treeing” in *idem*
 ZERO : See Section “Predefined Entities” in *lidoref*
 [] : See Section “Classes” in *lex*
 \040 : See Section “Quoting” in *lex*
 \040 : See Section “Skipping” in *lex*
 \b : See Section “Quoting” in *lex*
 \t : See Section “Quoting” in *lex*
 ^ : See Section “Classes” in *lex*
 acroread : See Section “Introduction” in *fw*
 auxCChar : See Section “Available scanners” in *lex*
 auxCChar : See Section “Definitions of Descriptions” in *lex*
 auxCComment : See Section “Available scanners” in *lex*
 auxCComment : See Section “Definitions of Descriptions” in *lex*
 auxCString : See Section “Available scanners” in *lex*
 auxCString : See Section “Definitions of Descriptions” in *lex*
 auxEOF : See Section “Available scanners” in *lex*
 auxEOL : See Section “Auxiliary Scanners” in *lex*
 auxEOL : See Section “Available scanners” in *lex*
 auxEOL : See Section “Definitions of Descriptions” in *lex*
 auxM2String : See Section “Available scanners” in *lex*
 auxM3Comment : See Section “Available scanners” in *lex*
 auxM3Comment : See Section “Definitions of Descriptions” in *lex*
 auxNUL : See Section “Available scanners” in *lex*
 auxNUL : See Section “auxNUL” in *news4.3*
 auxNewLine : See Section “Available scanners” in *lex*
 auxNoEOL : See Section “Available scanners” in *lex*
 auxNoEOL : See Section “auxNoEOL” in *news4.3*
 auxPascalComment : See Section “Available scanners” in *lex*
 auxPascalComment : See Section “Definitions of Descriptions” in *lex*
 auxPascalString : See Section “Available scanners” in *lex*
 auxPascalString : See Section “Definitions of Descriptions” in *lex*
 auxTab : See Section “Available scanners” in *lex*
 auxTab : See Section “Definitions of Descriptions” in *lex*
 c_mkchar : See Section “Available Processors” in *lex*
 c_mkchar : See Section “Definitions of Descriptions” in *lex*
 c_mkint : See Section “Available Processors” in *lex*
 c_mkint : See Section “Definitions of Descriptions” in *lex*
 c_mkstr : See Section “C_STRING_LIT” in *faq*
 c_mkstr : See Section “Available Processors” in *lex*
 computed trees : See Section “Bottom-up Evaluation” in *news4.1*
 coordAdjust : See Section “Available scanners” in *lex*
 dbx : See Section “debugex” in *novice*
 debug : See Section “debugex” in *novice*
 display : See Section “Introduction” in *fw*
 dvi : See Section “Introduction” in *fw*
 eli command : See Section “EliCmd” in *news4.6*

eli command : See Section “Cache” in *novice*
 eli command : See Section “CacheName” in *sysadmin*
 enter event : See Section “Supporting Profiling” in *mon*
 exe : See Section “Introduction” in *fw*
 exe : See Section “Products” in *novice*
 fwTexinfo : See Section “Introduction” in *fw*
 gdb : See Section “debugex” in *novice*
 gencode : See Section “Products” in *novice*
 gencode : See Section “debugex” in *novice*
 glalex : See Section “Generated Module” in *lex*
 help : See Section “Help” in *novice*
 help : See Section “Products” in *novice*
 help : See Section “parex” in *novice*
 idem : See Section “DeriveIdem” in *idem*
 idem : See Section “Unparsing” in *news4.1*
 installpkg product : See Section “installpkg” in *news4.6*
 inst-derivation : See Section “Instanciation of Modules with Filenames” in *news4.1*
 int : See Section “Predefined Entities” in *lidoref*
 leave event : See Section “Supporting Profiling” in *mon*
 lexerr : See Section “lexerr” in *news4.3*
 lint : See Section “debugex” in *novice*
 message : See Section “Instantiation” in *novice*
 mkidn : See Section “Combine” in *idem*
 mkidn : See Section “Available Processors” in *lex*
 mkidn : See Section “Canned Descriptions” in *lex*
 mkidn : See Section “Definitions of Descriptions” in *lex*
 mkidn : See Section “Folding” in *lex*
 mkidn : See Section “GLA specification” in *novice*
 mkint : See Section “Combine” in *idem*
 mkint : See Section “Available Processors” in *lex*
 mkint : See Section “Token Processors” in *lex*
 mkstr : See Section “C_STRING_LIT” in *faq*
 mkstr : See Section “Available Processors” in *lex*
 mkstr : See Section “Definitions of Descriptions” in *lex*
 modula_mkint : See Section “Available Processors” in *lex*
 modula_mkint : See Section “Definitions of Descriptions” in *lex*
 mondbx is no longer supported : See Section “Monitoring products” in *news4.3*
 mongdb : See Section “Invoking Noosa” in *mon*
 mongdb and +arg : See Section “Monitoring products” in *news4.3*
 mongdb is now working : See Section “Monitoring products” in *news4.3*
 mon : See Section “Invoking Noosa” in *mon*
 n: See [\[\]](#), page [\[\]](#) (mon)Database,,,aspects,aspects
 n: See [\[\]](#), page [\[\]](#) (mon)Database,,,events,events
 n_dectohex : See Section “Browsing non-standard types” in *mon*
 n_hextodec : See Section “Browsing non-standard types” in *mon*
 n_say_val : See Section “Browsing non-standard types” in *mon*
 n_say in browsing support : See Section “Browsing non-standard types” in *mon*

`n_say` in handlers : See Section “Breakpoints” in *mon*
`n_send` in browsing support : See Section “Browsing non-standard types” in *mon*
`n_send` to invoke operations : See Section “Breakpoints” in *mon*
`odin` command : See Section “Odin” in *sysadmin*
`parsable` : See Section “parex” in *novice*
`pdf` : See Section “Introduction” in *fw*
`return` : See Section “Operations” in *mon*
`return` statements : See Section “Operations” in *mon*
`source` : See Section “Products” in *novice*
`source` : See Section “symbex” in *novice*
`tcl` files : See Section “Browsing non-standard types” in *mon*
`tcl` files : See Section “User Initialisation” in *mon*
`test-utility` : See Section “ChangingFiles” in *faq*
`tree` : See Section “DeriveTree” in *idem*
`tree` : See Section “Unparsing” in *news4.1*
`typedef` : See Section “Declarations” in *tp*
`uname` command : See Section “CacheName” in *sysadmin*
`viewlist` : See Section “Products” in *novice*
`xdvi` : See Section “Introduction” in *fw*
`|` : See Section “Bar” in *lex*
`‘$elipkg/Scan/ScanProc.h’` : See Section “ScanProc” in *news4.3*
`‘.con’` file format : See Section “Grammars” in *syntax*
`‘.con’` file, example : See Section “Con” in *syntax*
`‘.con’` file, purpose : See Section “Notation” in *syntax*
`‘.dapto’` file format : See Section “Dapto Grammar” in *mon*
`‘.map’` file, example : See Section “Sym” in *syntax*
`‘.map’` file, format : See Section “Grammars” in *syntax*
`‘.noosarc’` : See Section “User Initialisation” in *mon*
`‘.perr’` file, format : See Section “Grammars” in *syntax*
`‘Add.fw’` : See Section “Use” in *idem*
`‘Analysis.fw’` : See Section “treelng” in *idem*
`‘C_Separator.fw’` : See Section “idemsep” in *idem*
`‘Cpp.fw’` : See Section “treelng” in *idem*
`‘ENV’` file : See Section “Implement” in *sysadmin*
`‘FINL.phi’` : See Section “Specifications” in *novice*
`‘HEAD.phi’` : See Section “Specifications” in *novice*
`‘INIT.phi’` : See Section “Specifications” in *novice*
`‘Idem.fw’` : See Section “treelng” in *idem*
`‘Java.fw’` : See Section “treelng” in *idem*
`‘M3.specs’` : See Section “DeriveTree” in *idem*
`‘M3.specs’` : See Section “treelng” in *idem*
`‘Odinfile’` : See Section “Session” in *novice*
`‘Odinfile’` : See Section “Odinfile” in *ui*
`‘PKGLST’` file : See Section “Capability” in *sysadmin*
`‘PKGLST’` file : See Section “Collections” in *sysadmin*
`‘SOCKET’` file : See Section “Communicate” in *sysadmin*
`‘SOCKET’` file : See Section “Implement” in *sysadmin*

`'Sp_Separator.fw'` : See Section “idemsep” in *idem*
`'Tree.fw'` : See Section “treelng” in *idem*
`'Xml.fw'` : See Section “treelng” in *idem*
`'clp'` : See Section “Specifications” in *novice*
`'con'` : See Section “Specifications” in *novice*
`'con'` file format : See Section “Grammars” in *syntax*
`'con'` file, example : See Section “Con” in *syntax*
`'con'` file, purpose : See Section “Notation” in *syntax*
`'ctl'` : See Section “Specifications” in *novice*
`'dapto'` file format : See Section “Dapto Grammar” in *mon*
`'delit'` : See Section “Specifications” in *novice*
`'eta'` : See Section “Specifications” in *novice*
`'example.fw'` : See Section “Use” in *idem*
`'fw'` : See Section “Generation” in *novice*
`'fw'` : See Section “Specifications” in *novice*
`'gla'` : See Section “Specifications” in *novice*
`'gnrc'` : See Section “Specifications” in *novice*
`'head'` : See Section “Output” in *novice*
`'lido'` : See Section “Specifications” in *novice*
`'map'` : See Section “Specifications” in *novice*
`'map'` file, example : See Section “Sym” in *syntax*
`'map'` file, format : See Section “Grammars” in *syntax*
`'oil'` : See Section “Specifications” in *novice*
`'pdl'` : See Section “Specifications” in *novice*
`'perr'` file, format : See Section “Grammars” in *syntax*
`'phi'` : See Section “Specifications” in *novice*
`'ptg'` : See Section “Specifications” in *novice*
`'specs'` : See Section “Specifications” in *novice*
`'str'` : See Section “Specifications” in *novice*
`'sym'` : See Section “Specifications” in *novice*
`$` symbol : See Section “Combine” in *idem*
`<` : See Section “Products” in *novice*
`<` : See Section “copy” in *ui*
`>` : See Section “Products” in *novice*
`>` : See Section “copy” in *ui*
`'+parser'` : See Section “Introduction” in *cola*
`'cola'` : See Section “Introduction” in *cola*
`'ERROR AUTOMATIC'` : See Section “Options” in *cola*
`'INFO'` : See Section “Options” in *cola*
`'OPT INFO'` : See Section “Options” in *cola*
`'colaOptInfo'` : See Section “Options” in *cola*
`'parsable'` : See Section “Options” in *cola*
`ODINPATH` environment variable : See Section “Capability” in *sysadmin*
`ODINPATH` environment variable : See Section “Odin” in *sysadmin*
`ODINVIEW` environment variable : See Section “CacheName” in *sysadmin*
`ODIN_LOCALIPC` : See Section “Implement” in *sysadmin*
`ODIN` environment variable : See Section “CacheName” in *sysadmin*

A Complete Example : See Section “Example” in *ptg*
 ACCESS : See Section “Operations” in *deftbl*
 ATTR : See Section “Value” in *comptrees*
 ATTREVAL : See Section “Refactor” in *news4.8*
 Abstract Data Types : See *adt*
 AdaptOil : See Section “ModLib3_6” in *oldmod*
 AddIdn : See Section “environment” in *lib*
 AddIdn : See Section “Bindings” in *name*
 AddKind : See Section “KindSet” in *prop*
 AddKindSet : See Section “KindSet” in *prop*
 AddToOrderedSet : See Section “List” in *adt*
 AddToOrderedSet : See Section “Adt” in *news4.3*
 AddToOrderedSetTYPEList : See Section “ModLib” in *news4.2*
 AddToSet : See Section “List” in *adt*
 AddToSetTYPEList : See Section “ModLib” in *news4.2*
 AddTypeToBlock : See Section “FuncType” in *type*
 AddTypeToBlock : See Section “Partition” in *type*
 AlgRangeSeq : See Section “Joined Ranges” in *name*
 AlgScope : See Section “Basic Scope Rules” in *name*
 Algol-like : See Section “Basic Scope Rules” in *name*
 Algol-like basic scope rules : See Section “AlgScope” in *name*
 Allocation : See Section “obstack” in *lib*
 AnyScope : See Section “AlgScope” in *name*
 AnyScope : See Section “CScope” in *name*
 App : See Section “List” in *adt*
 AppEl : See Section “List” in *adt*
 ArgumentDeListElem : See Section “Oil” in *news4.3*
 ArgumentDeListRoot : See Section “Oil” in *news4.3*
 ArgumentListElem : See Section “Oil” in *news4.3*
 ArgumentListRoot : See Section “Oil” in *news4.3*
 Arguments : See Section “Simple Expressions” in *lidoref*
 ArrayStorage : See Section “datamap” in *lib*
 AttrClass : See Section “Types and Classes of Attributes” in *lidoref*
 AttrName : See Section “Attributes” in *lidoref*
 AttrNames : See Section “Types and Classes of Attributes” in *lidoref*
 AttrSpecs : See Section “Types and Classes of Attributes” in *lidoref*
 Attribute : See Section “Attributes” in *lidoref*
 Attribute monitoring and browsing : See Section “Attributes” in *news4.3*
 Attributes : See Section “Attributes” in *lidoref*
 BNF : See Section “Notation” in *syntax*
 BOTTOMUP : See Section “Include” in *input*
 BOTTOMUP : See Section “Computations” in *lidoref*
 BOTTOMUP : See Section “Computed Subtrees” in *lidoref*
 BOTTOMUP : See Section “BuRangeSeq” in *name*
 BOTTOMUP : See Section “BuScope” in *name*
 BOTTOMUP : See Section “pgram” in *pp*
 Backus Naur Form : See Section “Notation” in *syntax*

BadIndication : See Section “Support” in *type*
 BadOperator : See Section “Support” in *type*
 BalanceContext : See Section “ExpressionSymbol” in *type*
 BalanceContext : See Section “Expressions” in *type*
 BalanceListElem : See Section “ExpressionSymbol” in *type*
 BalanceListElem : See Section “Expressions” in *type*
 BalanceListElem : See Section “OpndExprListRoot” in *type*
 BalanceListRoot : See Section “ExpressionSymbol” in *type*
 BalanceListRoot : See Section “Expressions” in *type*
 BalanceListRoot : See Section “OpndExprListRoot” in *type*
 Bind : See Section “AlgScope” in *name*
 Bind : See Section “Basic Scope Rules” in *name*
 Bind : See Section “CScope” in *name*
 BindIdn : See Section “Bindings” in *name*
 BindInScope : See Section “Bindings” in *name*
 BindKey : See Section “Bindings” in *name*
 BindKeyInScope : See Section “Bindings” in *name*
 Binding : See Section “AlgScope” in *name*
 Binding : See Section “Basic Scope Rules” in *name*
 Binding : See Section “CScope” in *name*
 Binding : See Section “Types” in *name*
 BindingInEnv : See Section “Lookup” in *name*
 BindingInScope : See Section “Lookup” in *name*
 Bool : See Section “ModLib3_6” in *oldmod*
 BottomUp : See Section “Pragmas” in *lidoref*
 BuRangeSeq : See Section “BuRangeSeq” in *name*
 BuRangeSeq : See Section “Joined Ranges” in *name*
 BuScope : See Section “Basic Scope Rules” in *name*
 BuScopeProp : See Section “Scope Properties” in *name*
 BuType : See Section “ModLib3_8” in *oldmod*
 BuildHosts : See Section “Client” in *sysadmin*
 C : See Section “CDeclarators” in *news4.4*
 C **return** statements : See Section “Operations” in *mon*
 C code : See Section “Actions” in *syntax*
 C compilation : See Section “cc” in *pp*
 C compilation : See Section “cc_flags” in *pp*
 C compilation : See Section “define” in *pp*
 C constant : See Section “Attribution” in *novice*
 C declarator : See Section “CDeclarators” in *news4.4*
 C literals : See Section “Simple Expressions” in *lidoref*
 C macro : See Section “Output” in *novice*
 C module : See Section “Introduction” in *ptg*
 C++ : See Section “Unparsers” in *news4.4*
 C-g : See Section “Help-Adv” in *info*
 C-l : See Section “Help-C-l” in *info*
 C-like : See Section “Basic Scope Rules” in *name*
 C-like basic scope Rules : See Section “CScope” in *name*

C-like basic scope rules : See Section “BuScope” in *name*
C-like inheritance bottom-up : See Section “BuInh” in *name*
C-p : See Section “Printing” in *info*
C-style comment : See Section “GLA specification” in *novice*
C-style comment : See Section “Specifications” in *novice*
C-t : See Section “Printing” in *info*
CHAIN : See Section “Basic” in *comptrees*
CHAIN : See Section “Chain” in *comptrees*
CHAIN : See Section “Remote” in *comptrees*
CHAIN : See Section “CHAIN” in *ptg*
CHAINSTART : See Section “Chain” in *comptrees*
CLASS SYMBOL : See Section “ModLib3.8” in *oldmod*
CLASS symbol : See Section “Inheritance” in *comptrees*
CLP_InputFile : See Section “Input parameter access” in *clp*
COLA : See Section “parser” in *pp*
COLA parser generating system : See Section “Phrase Structure” in *mon*
COMPUTE : See Section “Computations” in *lidoref*
CONSTITUENTS : See Section “Basic” in *comptrees*
CONSTITUENTS : See Section “Constituents” in *comptrees*
CONSTITUENTS : See Section “Remote” in *comptrees*
CONSTITUENTS : See Section “ModLib” in *news4.2*
CONSTITUENTS : See Section “CONSTITUENTS” in *ptg*
CPP : See Section “Implementing” in *comptrees*
CRangeSeq : See Section “Joined Ranges” in *name*
CScope : See Section “Basic Scope Rules” in *name*
CScopeProp : See Section “Scope Properties” in *name*
CType : See Section “ModLib3.8” in *oldmod*
CYCLE IN INDUCED GRAPHS : See Section “Using GORTO within ELI” in *gorto*
CYCLE IN PARTITIONED GRAPHS : See Section “Using GORTO within ELI” in *gorto*
C_Char : See Section “Simple Expressions” in *lidoref*
C_Float : See Section “Simple Expressions” in *lidoref*
C_IDENTIFIER_ISO : See Section “Available Descriptions” in *lex*
C_Integer : See Section “Simple Expressions” in *lidoref*
C_Name : See Section “Simple Expressions” in *lidoref*
C_Separator : See Section “Use” in *idem*
C_String : See Section “Simple Expressions” in *lidoref*
Cannot connect to Odin server : See Section “Connect” in *faq*
CastContext : See Section “ConversionContext” in *type*
CastContext : See Section “Expressions” in *type*
CatStrInd : See Section “Include” in *input*
CatStrInd : See Section “Message” in *problems*
CatStrInd : See Section “Strings” in *problems*
CatStrStr : See Section “Message” in *problems*
CatStrStr : See Section “Strings” in *problems*
Chain : See Section “ModLib3.6” in *oldmod*
ChainName : See Section “CHAIN” in *lidoref*
ChainPtg : See Section “ChainPtg obsolete” in *news4.1*

ChainPtg : See [Section “ModLib3_6”](#) in *oldmod*
CharPtr : See [Section “Strings”](#) in *problems*
ChkIdUse : See [Section “AlgScope”](#) in *name*
ChkIdUse : See [Section “Basic Scope Rules”](#) in *name*
ChkIdUse : See [Section “CScope”](#) in *name*
ChkInhIdUse : See [Section “AlgInh”](#) in *name*
ChkInhIdUse : See [Section “BuInh”](#) in *name*
ChkInhIdUse : See [Section “CInh”](#) in *name*
ChkInhIdUse : See [Section “Inheritance of Scopes”](#) in *name*
ChkInhIdUseScopeProp : See [Section “AlgInh”](#) in *name*
ChkInhIdUseScopeProp : See [Section “BuInh”](#) in *name*
ChkInhIdUseScopeProp : See [Section “CInh”](#) in *name*
ChkInhQualIdUse : See [Section “Inheritance of Scopes”](#) in *name*
ChkInherit : See [Section “AlgInh”](#) in *name*
ChkInherit : See [Section “BuInh”](#) in *name*
ChkInherit : See [Section “CInh”](#) in *name*
ChkInherit : See [Section “Inheritance of Scopes”](#) in *name*
ChkQualIdUse : See [Section “ScopeProp”](#) in *name*
ChkSetDeferId : See [Section “Defer”](#) in *prop*
ChkTypeDefDefId : See [Section “ChkTypeId”](#) in *type*
ChkTypeDefUseId : See [Section “ChkTypeId”](#) in *type*
ChkTypedUseId : See [Section “ChkTyped”](#) in *type*
ChkTypedUseId : See [Section “TypeIdDepend”](#) in *type*
CloneKey : See [Section “Keys”](#) in *deftbl*
CloneKey : See [Section “Definition table”](#) in *news4.3*
ClpOpenError : See [Section “Reporting open errors”](#) in *clp*
ClpOpenError : See [Section “Usage options”](#) in *clp*
ClpOpenError : See [Section “Command”](#) in *news4.8*
CmdLineIncl : See [Section “ModLib3_6”](#) in *oldmod*
CmpFctType : See [Section “List”](#) in *adt*
Coercible : See [Section “OpDef”](#) in *type*
ColOf : See [Section “error”](#) in *lib*
CombFctName : See [Section “CONSTITUENT\(S\)”](#) in *lidoref*
Comp : See [Section “List”](#) in *adt*
Computations : See [Section “Computations”](#) in *lidoref*
Computed Subtrees : See [Section “Computed Subtrees”](#) in *lidoref*
Concatenate : See [Section “datamap”](#) in *lib*
Cons : See [Section “List”](#) in *adt*
Continue command : See [Section “Controlling Your Program”](#) in *mon*
ConversionContext : See [Section “ConversionContext”](#) in *type*
ConversionContext : See [Section “Expressions”](#) in *type*
CoordLess : See [Section “SetFirst”](#) in *prop*
CoordMap : See [Section “ModLib3_6”](#) in *oldmod*
CoordPtr : See [Section “error”](#) in *lib*
Copy : See [Section “List”](#) in *adt*
CopyStorage : See [Section “datamap”](#) in *lib*
Count : See [Section “Counter”](#) in *problems*

Count : See Section “OccCnt” in *prop*
 CreateNewScope : See Section “BuInh” in *name*
 CreateNewScope : See Section “BuScope” in *name*
 CreateNewScope : See Section “BuScopeProp” in *name*
 CsmStrPtr : See Section “storage” in *lib*
 Csm_obstk : See Section “storage” in *lib*
 CurrTok : See Section “ModLib3_6” in *oldmod*
 Cygwin : See Section “Windows” in *news4.4*
 DEADLY : See Section “error” in *lib*
 DISPLAY : See Section “Using GORTO within ELI” in *gorto*
 DTD file : See Section “DeriveDefn” in *idem*
 DTD file : See Section “treelng” in *idem*
 DeListElem : See Section “LidoList” in *adt*
 DeListRoot : See Section “LidoList” in *adt*
 DeclaratorWithId : See Section “CScope” in *name*
 DefPt : See Section “ModLib3_6” in *oldmod*
 DefTableKey : See Section “Keys” in *deftbl*
 DefTableKeyList : See Section “Repeated options” in *clp*
 Defer : See Section “Defer” in *prop*
 Deferred Property Association : See Section “Defer” in *prop*
 DeferredKey : See Section “Defer” in *prop*
 DefineIdn : See Section “Keys” in *deftbl*
 DefineIdn : See Section “environment” in *lib*
 DefineIdn : See Section “Bindings” in *name*
 DefinitionsOf : See Section “Queries” in *name*
 DepAttr : See Section “Dependent Expressions” in *lidoref*
 DepAttrList : See Section “Dependent Expressions” in *lidoref*
 DepAttrs : See Section “Dependent Expressions” in *lidoref*
 Dependent Expressions : See Section “Dependent Expressions” in *lidoref*
 DependsClause : See Section “Dependent Expressions” in *lidoref*
 DescribeCoord command : See Section “Input Text Display” in *mon*
 DirectInherits : See Section “Queries” in *name*
 DyadicContext : See Section “Expressions” in *type*
 DyadicContext : See Section “OperatorSymbol” in *type*
 DyadicContext : See Section “Support” in *type*
 DyadicOperator : See Section “OpDef” in *type*
 DynSpace : See Section “ModLib3_6” in *oldmod*
 Dynamic Storage Allocation : See Section “DynSpace” in *adt*
 EDITOR : See Section “Options” in *info*
 ELI_DEBUGGER : See Section “Monitoring” in *pp*
 EOL markers, inserting : See Section “Inserting EOL” in *fw*
 EOL markers, suppressing : See Section “Suppressing End of Line” in *fw*
 ERROR : See Section “error” in *lib*
 Editing files in Noosa : See Section “File and handlers windows” in *news4.3*
 ElemIn : See Section “List” in *adt*
 Elements : See Section “Productions” in *lidoref*
 Eli : See Section “Specification” in *comptrees*

Eli package directory : See Section “EliInstall” in *sysadmin*
Eli session : See Section “Cache” in *novice*
Eli session : See Section “Session” in *novice*
Eli, installing : See Section “EliInstall” in *sysadmin*
Eli, typical use : See Section “Example” in *novice*
Emacs : See Section “Session” in *novice*
EndOfText : See Section “Available Processors” in *lex*
EnvOf : See Section “Queries” in *name*
EnvOfInherit : See Section “Queries” in *name*
Environment : See Section “Types” in *name*
Environment Module : See Section “Environment Module” in *name*
Environment Module : See Section “Environment Module Enhancements” in *news4.1*
Environment Module : See Section “Name” in *news4.3*
ErrorCount : See Section “error” in *lib*
ErrorInit : See Section “error” in *lib*
Event tracing dialog box : See Section “Main window command changes” in *news4.3*
Examples : See Section “Example” in *modlib*
Examples : See Section “top” in *modlib*
Examples : See Section “Basic Scope Rules” in *name*
Examples : See Section “top” in *name*
Examples, FunnelWeb : See Section “Introduction” in *fw*
ExpInfo : See Section “Attribute Class Conflict” in *comptrees*
ExpInfo : See Section “ExpInfo” in *pp*
ExportInhRange : See Section “AlgInh” in *name*
ExportInhRange : See Section “CInh” in *name*
ExportRange : See Section “AlgInh” in *name*
ExportRange : See Section “CScopeProp” in *name*
ExportRange : See Section “ScopeProp” in *name*
Expression : See Section “Dependent Expressions” in *lidoref*
ExpressionSymbol : See Section “AnalyzeExpression” in *type*
ExpressionSymbol : See Section “ChkExpr” in *type*
ExpressionSymbol : See Section “ExpressionSymbol” in *type*
ExpressionSymbol : See Section “Expressions” in *type*
ExpressionSymbol : See Section “FuncType” in *type*
ExpressionSymbol : See Section “OperatorSymbol” in *type*
ExpressionSymbol : See Section “Support” in *type*
Expressions : See Section “Expressions” in *lidoref*
Expressions : See Section “Expressions” in *ptg*
FAQ : See Section “top” in *faq*
Field : See Section “ModLib3.6” in *oldmod*
Filter : See Section “LidoList” in *adt*
FilterListElem : See Section “LidoList” in *adt*
FilterListElem : See Section “ModLib” in *news4.2*
FinalType : See Section “TypequDepend” in *type*
FindFile : See Section “CmdLineIncl” in *input*
Finl : See Section “List” in *adt*
FirstOcc : See Section “FirstOcc” in *prop*

ForEachStackElementDown : See Section “Stack” in *adt*
ForEachStackElementUp : See Section “Stack” in *adt*
Freq command : See Section “Frequency Profiles” in *mon*
FreqZero command : See Section “Frequency Profiles” in *mon*
Function Call Insertion : See Section “Calls” in *ptg*
FunctionName : See Section “Simple Expressions” in *lidoref*
FunnelWeb : See Section “FunnelWeb” in *news4.3*
FunnelWeb : See Section “PDF” in *news4.4*
FunnelWeb : See Section “Documentation” in *pp*
FunnelWeb Example : See Section “Introduction” in *fw*
FunnelWeb files : See Section “FunnelWeb” in *news4.1*
FunnelWeb typesetters : See Section “FunnelWeb” in *news4.2*
Fwd : See Section “ModLib3_6” in *oldmod*
GChain : See Section “ModLib3_6” in *oldmod*
GDB : See Section “Debugging” in *pp*
GLA : See Section “Tree Construction” in *comptrees*
GNU debugger : See Section “Debugging” in *pp*
GORTO.bak : See Section “Using GORTO within ELI” in *gorto*
GORTO.ctl : See Section “Using GORTO within ELI” in *gorto*
GenName : See Section “ModLib3_6” in *oldmod*
GenerateName : See Section “MakeName” in *problems*
Generating Identifiers : See Section “Identifiers” in *ptg*
Generating Output : See Section “top” in *output*
Get : See Section “GetCode” in *deftbl*
Get operations : See Section “Query” in *deftbl*
GetClpValue : See Section “Integer value options” in *clp*
GetClpValue : See Section “String value options” in *clp*
GetKind : See Section “Kind” in *prop*
GetKindSet : See Section “KindSet” in *prop*
GetScope : See Section “CScopeProp” in *name*
GetScope : See Section “ScopeProp” in *name*
GetScopeProp : See Section “BuScopeProp” in *name*
GotDeferProp : See Section “Defer” in *prop*
GotInhScopes : See Section “CInh” in *name*
GotOper : See Section “FuncType” in *type*
GotOper : See Section “OilClass” in *type*
GotOper : See Section “OpDef” in *type*
GotScopes : See Section “ModLib3_8” in *oldmod*
GotScopesDefs : See Section “ModLib3_8” in *oldmod*
GotType : See Section “FuncType” in *type*
GotType : See Section “OpDef” in *type*
GotType : See Section “TypeDenot” in *type*
GotVisibleKeys : See Section “AlgInh” in *name*
GotVisibleKeysNest : See Section “AlgInh” in *name*
HEAD : See Section “Chain” in *comptrees*
Handlers command : See Section “Breakpoints” in *mon*
Has : See Section “Has” in *deftbl*

Has : See Section “Has” in *news4.4*
Hash : See Section “Hashing” in *news4.4*
Head : See Section “List” in *adt*
Hiding of attributes : See Section “The Main Window” in *gorto*
INCLUDING : See Section “Basic” in *comptrees*
INCLUDING : See Section “Including” in *comptrees*
INCLUDING : See Section “Remote” in *comptrees*
INFORUNDISPLAY : See Section “Options” in *info*
INH : See Section “Basic” in *comptrees*
INH : See Section “Value” in *comptrees*
INHERITS : See Section “Inheritance” in *comptrees*
IPC not available : See Section “IPC” in *faq*
IdDef : See Section “ModLib3_8” in *oldmod*
IdDefScope : See Section “AlgScope” in *name*
IdDefScope : See Section “Basic Scope Rules” in *name*
IdDefScope : See Section “CScope” in *name*
IdDefScopeProp : See Section “ModLib3_8” in *oldmod*
IdDefUse : See Section “CScope” in *name*
IdGetScopeProp : See Section “ModLib3_8” in *oldmod*
IdInDeclarator : See Section “CScope” in *name*
IdSetScopeProp : See Section “BuScopeProp” in *name*
IdUse : See Section “ModLib3_8” in *oldmod*
IdUseEnv : See Section “AlgScope” in *name*
IdUseEnv : See Section “Basic Scope Rules” in *name*
IdUseEnv : See Section “CScope” in *name*
IdUseEnv : See Section “Inheritance of Scopes” in *name*
IdUseScope : See Section “AlgScope” in *name*
IdUseScope : See Section “Basic Scope Rules” in *name*
IdUseScope : See Section “CScope” in *name*
IdUseScope : See Section “Inheritance of Scopes” in *name*
IdUseScope : See Section “ModLib3_8” in *oldmod*
Idem and literal chain rules : See Section “Completion” in *syntax*
IdnNumb : See Section “MakeName” in *problems*
IdnOf : See Section “Queries” in *name*
InclLido : See Section “ModLib3_6” in *oldmod*
Include : See Section “ModLib3_6” in *oldmod*
IndCatStrStr : See Section “Strings” in *problems*
Indent : See Section “ModLib3_6” in *oldmod*
Indexed Insertion Points : See Section “Indexed” in *ptg*
Indication : See Section “ConversionContext” in *type*
Indication : See Section “Expressions” in *type*
Indication : See Section “FuncType” in *type*
Indication : See Section “Support” in *type*
InhPrecond : See Section “BuInh” in *name*
InhRange : See Section “AlgInh” in *name*
InhRange : See Section “CInh” in *name*
InhRange : See Section “Inheritance of Scopes” in *name*

InheritClass : See Section “Inheritance” in *name*
 InheritClass : See Section “Inheritance of Scopes” in *name*
 InheritOk : See Section “Inheritance of Scopes” in *name*
 InheritPtr : See Section “Types” in *name*
 InheritScope : See Section “AlgInh” in *name*
 InheritScope : See Section “BuInh” in *name*
 InheritScope : See Section “CInh” in *name*
 InheritScope : See Section “Inheritance of Scopes” in *name*
 InheritScopeProp : See Section “ModLib3_8” in *oldmod*
 Inheritance : See Section “Inheritance of Computations” in *lidoref*
 Inheritance : See Section “Symbol Specifications” in *lidoref*
 Inheritance of Computations : See Section “Inheritance of Computations” in *lidoref*
 Inheritsfrom : See Section “Inheritance” in *name*
 InnerScope : See Section “Inheritance of Scopes” in *name*
 Input Processing : See Section “top” in *input*
 InsertAfter : See Section “List” in *adt*
 InstClass : See Section “OilClass” in *type*
 InstClass1 : See Section “OilClass” in *type*
 IntSet : See Section “ModLib3_6” in *oldmod*
 IntSet : See Section “KindSet” in *prop*
 IntToKey : See Section “ModLib3_6” in *oldmod*
 IntToPtr : See Section “ModLib3_6” in *oldmod*
 Introduction : See Section “Introduction” in *lidoref*
 Introduction : See Section “Introduction” in *ptg*
 Is : See Section “Is” in *deftbl*
 Is : See Section “IsCode” in *deftbl*
 IsClass : See Section “Queries” in *name*
 IsCoercible : See Section “Support” in *type*
 IsName : See Section “Is” in *deftbl*
 IsType : See Section “Types” in *type*
 Iteration : See Section “Iterations” in *lidoref*
 Iterations : See Section “Iterations” in *lidoref*
 IthElem : See Section “List” in *adt*
 Java : See Section “Unparsers” in *news4.4*
 Joined Ranges : See Section “Joined Ranges” in *name*
 KReset : See Section “PropLib” in *prop*
 KeyInEnv : See Section “environment” in *lib*
 KeyInEnv : See Section “Lookup” in *name*
 KeyInRange : See Section “environment” in *lib*
 KeyInScope : See Section “Lookup” in *name*
 KeyOf : See Section “Queries” in *name*
 KeyOfEnv : See Section “Queries” in *name*
 Kill command : See Section “Controlling Your Program” in *mon*
 Kind : See Section “Kind” in *prop*
 KindSet : See Section “ModLib3_6” in *oldmod*
 KindSet : See Section “KindSet” in *prop*
 LALR: See Section “condition (novice)parex” in *1*

LIDO : See Section “Introduction” in *lidoref*
LIDO : See Section “CHAIN” in *ptg*
LIDO : See Section “CONSTITUENTS” in *ptg*
LIDO : See Section “Example” in *ptg*
LIDO : See Section “Expressions” in *ptg*
LIGA : See Section “Introduction” in *lidoref*
Leaf Pattern : See Section “Outdated” in *ptg*
LeafPtg : See Section “ModLib3_6” in *oldmod*
Length : See Section “List” in *adt*
Library : See Section “IncludeFiles” in *faq*
Library : See Section “top” in *modlib*
Library Adt : See *adt*
Library Input : See Section “top” in *input*
Library Name : See Section “top” in *name*
Library Output : See Section “top” in *output*
Library Prop : See Section “top” in *prop*
Library Tech : See Section “top” in *problems*
LidoList-Module : See Section “ModLib” in *news4.2*
Line : See Section “Name Analysis Test” in *name*
LineNum : See Section “error” in *lib*
LineOf : See Section “error” in *lib*
Lines in a FunnelWeb file : See Section “FunnelWeb” in *news4.3*
List : See Section “List” in *adt*
List : See Section “ModLib3_6” in *oldmod*
List-Module : See Section “ModLib” in *news4.2*
ListContext : See Section “Expressions” in *type*
ListContext : See Section “FuncType” in *type*
ListContext : See Section “OpDef” in *type*
ListContext : See Section “OpndExprListRoot” in *type*
ListContext : See Section “Support” in *type*
ListElem : See Section “LidoList” in *adt*
ListElem : See Section “ModLib” in *news4.2*
ListOperator : See Section “FuncType” in *type*
ListOperator : See Section “OpDef” in *type*
ListPtr : See Section “List” in *adt*
ListRoot : See Section “LidoList” in *adt*
MAPCHAINS keyword : See Section “Completion” in *syntax*
MAPCHAINS keyword : See Section “Mapping Chain Rules” in *syntax*
MAPRULE keyword : See Section “Rule Mapping” in *syntax*
MAPSYM keyword : See Section “Symbol Mapping” in *syntax*
Main Menu : See Section “The Main Window” in *gorto*
Main Window : See Section “The Main Window” in *gorto*
MakeName : See Section “MakeName” in *problems*
Map : See Section “List” in *adt*
MapFct : See Section “List” in *adt*
Maptool : See Section “Tree Construction” in *comptrees*
Maptool : See *syntax*

Message : See [Section “ModLib3_6”](#) in *oldmod*
Mk functions : See [Section “nomain”](#) in *news4.4*
Mk-Functions : See [Section “Tree Construction Functions”](#) in *lidoref*
Modification of Partitions : See [Section “Symbols”](#) in *gorto*
Modification visit-sequences : See [Section “Visit-sequences”](#) in *gorto*
Module : See [Section “IncludeFiles”](#) in *faq*
Module AlgInh : See [Section “AlgInh”](#) in *name*
Module AlgRangeSeq : See [Section “AlgRangeSeq”](#) in *name*
Module AlgScope : See [Section “AlgScope”](#) in *name*
Module BitSet : See [Section “BitSet”](#) in *adt*
Module BuInh : See [Section “BuInh”](#) in *name*
Module BuRangeSeq : See [Section “BuRangeSeq”](#) in *name*
Module BuScopeProp : See [Section “BuScopeProp”](#) in *name*
Module BuScope : See [Section “BuScope”](#) in *name*
Module BuType : See [Section “ModLib3_8”](#) in *oldmod*
Module CInh : See [Section “CInh”](#) in *name*
Module CRangeSeq : See [Section “CRangeSeq”](#) in *name*
Module CScopeProp : See [Section “CScopeProp”](#) in *name*
Module CScope : See [Section “CScope”](#) in *name*
Module CType : See [Section “ModLib3_8”](#) in *oldmod*
Module CmdLineIncl : See [Section “CmdLineIncl”](#) in *input*
Module CoordMap : See [Section “top”](#) in *input*
Module Counter : See [Section “Counter”](#) in *problems*
Module CurrTok : See [Section “CurrTok”](#) in *input*
Module DynSpace : See [Section “DynSpace”](#) in *adt*
Module FirstOcc : See [Section “FirstOcc”](#) in *prop*
Module GlaCorr : See [Section “top”](#) in *input*
Module Hash : See [Section “Hash”](#) in *problems*
Module Include : See [Section “Include”](#) in *input*
Module Indent : See [Section “Indent”](#) in *output*
Module IntSet : See [Section “IntSet”](#) in *adt*
Module IntSet : See [Section “KindSet”](#) in *prop*
Module KindSet : See [Section “KindSet”](#) in *prop*
Module Kind : See [Section “Kind”](#) in *prop*
Module LeafPtg : See [Section “LeafPtg”](#) in *output*
Module LidoList : See [Section “LidoList”](#) in *adt*
Module LidoList : See [Section “ModLib”](#) in *news4.2*
Module List : See [Section “List”](#) in *adt*
Module List : See [Section “ModLib”](#) in *news4.2*
Module MakeName : See [Section “MakeName”](#) in *problems*
Module Map : See [Section “Map”](#) in *adt*
Module Message : See [Section “Message”](#) in *problems*
Module ObjCnt : See [Section “ObjCnt”](#) in *prop*
Module OccCnt : See [Section “OccCnt”](#) in *prop*
Module OutStr : See [Section “OutStr”](#) in *output*
Module PreDefId : See [Section “Predefined Identifiers”](#) in *name*
Module PreDefine : See [Section “Predefined Identifiers”](#) in *name*

Module PropLib : See Section “PropLib” in *prop*
Module PtgCommon : See Section “PtgCommon” in *output*
Module Reflex : See Section “Reflex” in *prop*
Module ScopeProp : See Section “ScopeProp” in *name*
Module SetFirst : See Section “SetFirst” in *prop*
Module ShowBinding : See Section “Name Analysis Test” in *name*
Module Sort : See Section “Sort” in *problems*
Module Stack : See Section “Stack” in *adt*
Module StrArith : See Section “StrArith” in *problems*
Module Table : See Section “Table” in *adt*
Module Type : See Section “ModLib3_8” in *oldmod*
Module Unique : See Section “Unique” in *prop*
Module BitSet : See Section “ModLib3_8” in *oldmod*
Module Defer : See Section “Defer” in *prop*
Module DynSpace : See Section “ModLib3_8” in *oldmod*
Module LeafPtg : See Section “ModLib3_8” in *oldmod*
Module List : See Section “ModLib3_8” in *oldmod*
Module PreDefMod : See Section “Predefined Identifiers” in *name*
Module PropLib : See Section “ModLib3_8” in *oldmod*
Module PropLib : See Section “Defer” in *prop*
Module PtrList : See Section “List” in *adt*
Module envmod : See Section “Environment Module” in *name*
Modules for Name Analysis : See Section “ModLib3_8” in *oldmod*
MonadicContext : See Section “Expressions” in *type*
MonadicContext : See Section “OperatorSymbol” in *type*
MonadicContext : See Section “OpndExprListRoot” in *type*
MonadicOperator : See Section “OpDef” in *type*
Monitoring : See Section “Monitoring” in *news4.1*
Monitoring : See Section “Monitoring” in *news4.3*
Multiple +arg parameters : See Section “Monitoring products” in *news4.3*
NONTERM : See Section “LIGA” in *news4.5*
NOTE : See Section “error” in *lib*
NUL : See Section “source” in *lib*
NULLIS : See Section “IntSet” in *adt*
NULLTYPEList : See Section “List” in *adt*
Name equivalence : See Section “UserTypes” in *type*
Names : See Section “Overall Structure” in *lidoref*
Nest : See Section “ModLib3_6” in *oldmod*
NewEnv : See Section “environment” in *lib*
NewEnv : See Section “Constructors” in *name*
NewInput : See Section “CurrTok” in *input*
NewInput : See Section “Include” in *input*
NewKey : See Section “Interface” in *deftbl*
NewKey : See Section “Keys” in *deftbl*
NewScope : See Section “environment” in *lib*
NewScope : See Section “BuInh” in *name*
NewScope : See Section “Constructors” in *name*

NewStorage : See [Section “datamap”](#) in *lib*
NextDefinition : See [Section “Queries”](#) in *name*
NextInhBinding : See [Section “Hidden”](#) in *name*
NextInhKey : See [Section “Hidden”](#) in *name*
NextInherit : See [Section “Queries”](#) in *name*
NoBinding : See [Section “Types”](#) in *name*
NoCoord : See [Section “error”](#) in *lib*
NoEnv : See [Section “Types”](#) in *name*
NoIdn : See [Section “identifier”](#) in *lib*
NoInherit : See [Section “Types”](#) in *name*
NoKey : See [Section “Integer value options”](#) in *clp*
NoKey : See [Section “Interface”](#) in *deftbl*
NoKey : See [Section “Keys”](#) in *deftbl*
NoKey : See [Section “Query”](#) in *deftbl*
NoKey : See [Section “Update”](#) in *deftbl*
NoKeyMsg : See [Section “ModLib3-6”](#) in *oldmod*
NoPosition : See [Section “error”](#) in *lib*
NoStorage : See [Section “datamap”](#) in *lib*
NoStr : See [Section “storage”](#) in *lib*
NoStrIndex : See [Section “storage”](#) in *lib*
Node command : See [Section “Trees”](#) in *news4.3*
Noosa : See [Section “top”](#) in *mon*
Noosa : See [Section “Monitoring”](#) in *news4.1*
Noosa : See [Section “Monitoring”](#) in *news4.3*
Noosa : See [Section “Monitoring”](#) in *pp*
Noosa : See [Section “monitor”](#) in *pp*
Noosa X resources : See [Section “Configuring Noosa”](#) in *news4.3*
Noosa handlers : See [Section “File and handlers windows”](#) in *news4.3*
Noosa keyboard accelerators : See [Section “Main window command changes”](#) in *news4.3*
Noosa searching and saving : See [Section “Main window command changes”](#) in *news4.3*
Noosa tearoff menus : See [Section “Main window command changes”](#) in *news4.3*
Noosa tree displays : See [Section “Trees”](#) in *news4.3*
Noosa.treeCompHeight : See [Section “X resources”](#) in *mon*
Noosa.treeCompWidth : See [Section “X resources”](#) in *mon*
Noosa.treeFullHeight : See [Section “X resources”](#) in *mon*
Noosa.treeFullWidth : See [Section “X resources”](#) in *mon*
Noosa.treeIncrHeight : See [Section “X resources”](#) in *mon*
Noosa.treeIncrWidth : See [Section “X resources”](#) in *mon*
Noosa.treeSrcHeight : See [Section “X resources”](#) in *mon*
Noosa.treeSrcWidth : See [Section “X resources”](#) in *mon*
NullBitSet : See [Section “BitSet”](#) in *adt*
NullFctName : See [Section “CONSTITUENT\(S\)”](#) in *lidoref*
NullTYPEList : See [Section “List”](#) in *adt*
Number : See [Section “Attributes”](#) in *lidoref*
OIL CLASS, definition : See [Section “OilClass”](#) in *type*
OIL CLASS, instantiation : See [Section “OilClass”](#) in *type*
OIL COERCION : See [Section “Coercibility”](#) in *type*

OIL COERCION : See Section “OilClass” in *type*
OIL INDICATION : See Section “Indications” in *type*
OIL INDICATION : See Section “OilClass” in *type*
OIL OPER : See Section “OilClass” in *type*
OIL OPER : See Section “Operators” in *type*
OIL comments : See Section “Comments” in *oil*
OIL identifiers : See Section “Identifiers” in *oil*
OIL library : See Section “Library” in *oil*
OIL specification : See Section “Language” in *oil*
OIL statements : See Section “Language” in *oil*
OIL statements : See Section “Statements” in *oil*
ObjCnt : See Section “ObjCnt” in *prop*
OccCnt : See Section “ModLib3_6” in *oldmod*
Odin : See Section “Package” in *news4.6*
Odin client : See Section “Execution” in *sysadmin*
Odin execution : See Section “Execution” in *sysadmin*
Odin package directory : See Section “EliInstall” in *sysadmin*
Odin server : See Section “Execution” in *sysadmin*
Odin variables : See Section “Odinrc” in *faq*
OilAddArgSig : See Section “Signature Constructor” in *oil*
OilAddClassCoercion : See Section “Class Constructor” in *oil*
OilAddClassOp : See Section “Class Constructor” in *oil*
OilAddClassOpId : See Section “Class Constructor” in *oil*
OilAddClassSigArg : See Section “Class Constructor” in *oil*
OilAddCoercion : See Section “Coercion Constructor” in *oil*
OilAddIdentification : See Section “Identification Constructor” in *oil*
OilBalance : See Section “Possible Types” in *oil*
OilClassInst0 : See Section “Instantiate Classes” in *oil*
OilClassInst1 : See Section “Instantiate Classes” in *oil*
OilClassInst2 : See Section “Instantiate Classes” in *oil*
OilClassName : See Section “Name Functions” in *oil*
OilCoerce : See Section “Coercion Sequences” in *oil*
OilEmptyCS : See Section “Coercion Sequences” in *oil*
OilGetArg : See Section “Simple Example” in *oil*
OilGetArgType : See Section “Signature” in *oil*
OilHeadCS : See Section “Coercion Sequences” in *oil*
OilIdOp1 : See Section “Simplification” in *oil*
OilIdOp2 : See Section “Simplification” in *oil*
OilIdOp3 : See Section “Simplification” in *oil*
OilIdOpTS1 : See Section “Possible Types” in *oil*
OilIdOpTS2 : See Section “Possible Types” in *oil*
OilIdOpTS3 : See Section “Possible Types” in *oil*
OilIdResultTS1 : See Section “Possible Types” in *oil*
OilIdResultTS2 : See Section “Possible Types” in *oil*
OilIdResultTS3 : See Section “Possible Types” in *oil*
OilIsValidCS : See Section “Coercion Sequences” in *oil*
OilIsValidOp : See Section “Validation” in *oil*

OilNewArgSig : See Section “Signature Constructor” in *oil*
 OilNewClass : See Section “Class Constructor” in *oil*
 OilNewClassSigArg : See Section “Class Constructor” in *oil*
 OilNewOp : See Section “Operator Constructor” in *oil*
 OilNewType : See Section “Type Constructor” in *oil*
 OilNoOverload : See Section “Possible Types” in *oil*
 OilOpName : See Section “Name Functions” in *oil*
 OilSelectTypeFromTS : See Section “Possible Types” in *oil*
 OilSetIncludes : See Section “Possible Types” in *oil*
 OilTailCS : See Section “Coercion Sequences” in *oil*
 OilTypeName : See Section “Name Functions” in *oil*
 OilTypeToSet : See Section “Possible Types” in *oil*
 Once : See Section “ModLib3_6” in *oldmod*
 OpenElemScope : See Section “BuRangeSeq” in *name*
 OpenNewScope : See Section “BuScope” in *name*
 OpenNewScope : See Section “BuScopeProp” in *name*
 OpenPrecond : See Section “BuScopeProp” in *name*
 OpenSeqScope : See Section “BuRangeSeq” in *name*
 OperName : See Section “Expressions” in *type*
 Operator-Module : See Section “Instanciation of Modules with Filenames” in *news4.1*
 OperatorDefs : See Section “FuncType” in *type*
 OperatorDefs : See Section “OilClass” in *type*
 OperatorDefs : See Section “OpDef” in *type*
 OperatorSymbol : See Section “ChkExpr” in *type*
 OperatorSymbol : See Section “Expressions” in *type*
 OperatorSymbol : See Section “OpDef” in *type*
 OperatorSymbol : See Section “OperatorSymbol” in *type*
 OpndExprListElem : See Section “ChkExpr” in *type*
 OpndExprListElem : See Section “ConversionContext” in *type*
 OpndExprListElem : See Section “Expressions” in *type*
 OpndExprListElem : See Section “OpndExprListRoot” in *type*
 OpndExprListRoot : See Section “ChkExpr” in *type*
 OpndExprListRoot : See Section “Expressions” in *type*
 OpndExprListRoot : See Section “FuncType” in *type*
 OpndExprListRoot : See Section “OpDef” in *type*
 OpndExprListRoot : See Section “OpndExprListRoot” in *type*
 OpndExprListRoot : See Section “Support” in *type*
 OpndTypeList : See Section “OpDef” in *type*
 OpndTypeListElem : See Section “OpDef” in *type*
 OpndTypeListRoot : See Section “FuncType” in *type*
 OpndTypeListRoot : See Section “OpDef” in *type*
 OptimInfo : See Section “OptimInfo” in *pp*
 OrdInfo : See Section “OrdInfo” in *pp*
 OrderedInsert : See Section “List” in *adt*
 OutStr : See Section “ModLib3_6” in *oldmod*
 Outdated Constructs : See Section “Outdated Constructs” in *lidoref*
 Outdated Constructs : See Section “Outdated” in *ptg*

OuterScope : See Section “Inheritance of Scopes” in *name*
Output Functions : See Section “Output” in *ptg*
Output of Data Items : See Section “Data” in *ptg*
Output of Sequences : See Section “Sequences” in *ptg*
Overall Structure : See Section “Overall Structure” in *lidoref*
Overlay : See Section “datamap” in *lib*
OverridesBinding : See Section “Hidden” in *name*
PASCAL : See Section “Example” in *oil*
PASCAL : See Section “Simple Example” in *oil*
PDF files : See Section “PDF” in *news4.4*
PDL : See Section “top” in *prop*
PDL pattern : See Section “PropLib” in *prop*
PGS : See Section “parser” in *pp*
PGS parser generating system : See Section “Phrase Structure” in *mon*
PRESENT : See Section “Operations” in *deftbl*
PRINTER : See Section “Options” in *info*
PTG : See *ptg*
PTG Output : See Section “StringOut” in *output*
PTG patterns, overriding : See Section “Changing” in *idem*
PTG specification : See Section “Output” in *novice*
PTGFree : See Section “Predefined” in *ptg*
PTGNULL : See Section “Predefined” in *ptg*
PTGNode : See Section “Output” in *ptg*
PTGNode : See Section “Patterns” in *ptg*
PTGNode : See Section “Predefined” in *ptg*
PTGNode : See Section “Typed” in *ptg*
PTGNull : See Section “Predefined” in *ptg*
PTGOut : See Section “Output” in *ptg*
PTGOut : See Section “Predefined” in *ptg*
PTGOutFPtr : See Section “Output” in *ptg*
PTGOutFPtr : See Section “Predefined” in *ptg*
PTGOutFile : See Section “Output” in *ptg*
PTGOutFile : See Section “Predefined” in *ptg*
PTGProcess : See Section “Predefined” in *ptg*
Packages : See Section “Packages” in *sysadmin*
ParameterListElem : See Section “Oil” in *news4.3*
ParameterListRoot : See Section “Oil” in *news4.3*
ParameterTypeList : See Section “FuncType” in *type*
ParentOf : See Section “Queries” in *name*
Pascal : See Section “SetFirst” in *prop*
Pattern Specifications : See Section “Patterns” in *ptg*
Pattern-Based Text Generator : See *ptg*
Phrase command : See Section “Phrase Structure” in *mon*
PostDeListElem : See Section “LidoList” in *adt*
PostListElem : See Section “LidoList” in *adt*
Postscript output from tree displays : See Section “Trees” in *news4.3*
PreDeListElem : See Section “LidoList” in *adt*

PreDef : See [Section “ModLib3_6”](#) in *oldmod*
 PreDefBind : See [Section “Predefined Identifiers”](#) in *name*
 PreDefId-Module : See [Section “Instanciation of Modules with Filenames”](#) in *news4.1*
 PreDefKey : See [Section “Predefined Identifiers”](#) in *name*
 PreDefKeyBind : See [Section “Predefined Identifiers”](#) in *name*
 PreDefMod : See [Section “Predefined Identifiers”](#) in *name*
 PreDefMod : See [Section “Name”](#) in *news4.3*
 PreDefSym : See [Section “Predefined Identifiers”](#) in *name*
 PreDefSymKey : See [Section “Predefined Identifiers”](#) in *name*
 PreDefSymKeyBind : See [Section “Predefined Identifiers”](#) in *name*
 PreDefine : See [Section “Predefined Identifiers”](#) in *name*
 PreDefine : See [Section “Name”](#) in *news4.3*
 PreDefineSym : See [Section “Predefined Identifiers”](#) in *name*
 PreDefineSym : See [Section “Name”](#) in *news4.3*
 PreIdnPost : See [Section “MakeName”](#) in *problems*
 PreListElem : See [Section “LidoList”](#) in *adt*
 Predefined Entities : See [Section “Predefined Entities”](#) in *lidoref*
 Predefined Entities : See [Section “Predefined”](#) in *ptg*
 Predefined Entities : See [Section “Predefined”](#) in *tp*
 PrimaryContext : See [Section “ExpressionSymbol”](#) in *type*
 PrimaryContext : See [Section “Expressions”](#) in *type*
 PrimaryContext : See [Section “TypeIdDepend”](#) in *type*
 Production : See [Section “Productions”](#) in *lidoref*
 Production : See [Section “Rule Specifications”](#) in *lidoref*
 Production Menu : See [Section “Productions”](#) in *gorto*
 Productions : See [Section “Productions”](#) in *lidoref*
 Program Frame : See [Section “Frame”](#) in *ptg*
 Prop : See [Section “top”](#) in *prop*
 PropLib : See [Section “Defer”](#) in *prop*
 PropLib : See [Section “SetFirst”](#) in *prop*
 Ptg-Pattern AsIs : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern CChar : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern CString : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern CommaSeq : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern Eol : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern Id : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern Numb : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern PString : See [Section “Frequently Used Patterns”](#) in *output*
 Ptg-Pattern Seq : See [Section “Frequently Used Patterns”](#) in *output*
 PtrList : See [Section “List”](#) in *adt*
 PtrList : See [Section “ModLib3_6”](#) in *oldmod*
 QualIdUse : See [Section “AlgInh”](#) in *name*
 QualIdUse : See [Section “BuScopeProp”](#) in *name*
 QualIdUse : See [Section “CInh”](#) in *name*
 QualIdUse : See [Section “CScopeProp”](#) in *name*
 QualIdUse : See [Section “Inheritance of Scopes”](#) in *name*
 QualIdUse : See [Section “ScopeProp”](#) in *name*

R parameter of the eli command : See Section “Upgrading” in *faq*
RHS : See Section “Attributes” in *lidoref*
ROOTCLASS : See Section “ModLib” in *news4.1*
RULE : See Section “Value” in *comptrees*
RangeCnt : See Section “OccCnt” in *prop*
RangeElement : See Section “Joined Ranges” in *name*
RangeFirstOcc : See Section “FirstOcc” in *prop*
RangeKind : See Section “Kind” in *prop*
RangeKindSet : See Section “KindSet” in *prop*
RangeObjCnt : See Section “ObjCnt” in *prop*
RangeScope : See Section “AlgScope” in *name*
RangeScope : See Section “Basic Scope Rules” in *name*
RangeScope : See Section “BuInh” in *name*
RangeScope : See Section “CScope” in *name*
RangeScopeProp : See Section “ModLib3-8” in *oldmod*
RangeSequence : See Section “BuRangeSeq” in *name*
RangeSequence : See Section “Joined Ranges” in *name*
RangeSetFirst : See Section “SetFirst” in *prop*
RangeUnique : See Section “Unique” in *prop*
RecentNewScope : See Section “BuInh” in *name*
RecentNewScope : See Section “BuScope” in *name*
RefEndApp : See Section “List” in *adt*
RefEndCons : See Section “List” in *adt*
ReflexFrom : See Section “Reflex” in *prop*
ReflexTo : See Section “Reflex” in *prop*
RemAttrList : See Section “INCLUDING” in *lidoref*
RemAttrs : See Section “INCLUDING” in *lidoref*
Remote Attribute Access : See Section “Remote Attribute Access” in *lidoref*
RemoteAccess : See Section “CONSTITUENT(S)” in *lidoref*
RemoteAccess : See Section “INCLUDING” in *lidoref*
Reparatur : See Section “Error Recovery” in *syntax*
Reset : See Section “ResetCode” in *deftbl*
Reset operation : See Section “Update” in *deftbl*
RhsAttrs : See Section “Attributes” in *lidoref*
RhsAttrs : See Section “Pragmas” in *lidoref*
RhsFct : See Section “Predefined Entities” in *lidoref*
RootContext : See Section “ConversionContext” in *type*
RootContext : See Section “Expressions” in *type*
RootCount : See Section “Counter” in *problems*
RootDefer : See Section “Defer” in *prop*
RootEnv : See Section “AlgScope” in *name*
RootEnv : See Section “CScope” in *name*
RootEnv : See Section “Predefined Identifiers” in *name*
RootScope : See Section “AlgScope” in *name*
RootScope : See Section “Basic Scope Rules” in *name*
RootScope : See Section “CScope” in *name*
Rule Mapping : See Section “Rule Mapping” in *syntax*

Rule Specifications : See Section “Rule Specifications” in *lidoref*
 RuleAttr : See Section “Attributes” in *lidoref*
 RuleFct : See Section “Attributes” in *lidoref*
 RuleFct : See Section “Pragmas” in *lidoref*
 RuleFct : See Section “Predefined Entities” in *lidoref*
 Run command : See Section “Controlling Your Program” in *mon*
 SHIELD clause : See Section “Constituents” in *comptrees*
 SRCFILE : See Section “source” in *lib*
 SRCFILE : See Section “Frame” in *ptg*
 STOPAFTERBADPARSE : See Section “define” in *pp*
 STOPAFTERBADPARSE : See Section “Error Recovery” in *syntax*
 SYMBOL : See Section “Basic” in *comptrees*
 SYMBOL : See Section “Value” in *comptrees*
 SYNT : See Section “Basic” in *comptrees*
 SYNT : See Section “Value” in *comptrees*
 Scope : See Section “Inheritance of Scopes” in *name*
 ScopeKey : See Section “BuScopeProp” in *name*
 ScopeKey : See Section “Inheritance of Scopes” in *name*
 ScopeProp : See Section “Scope Properties” in *name*
 Searching and saving Noosa text : See Section “Main window command changes” in *news4.3*
 Separator : See Section “Separator added” in *news4.1*
 Separator : See Section “Separator” in *output*
 Separator function call : See Section “Use” in *idem*
 Separator function call : See Section “idemsep” in *idem*
 Set : See Section “SetCode” in *deftbl*
 Set operations : See Section “Update” in *deftbl*
 SetDeferId : See Section “Defer” in *prop*
 SetDeferProp : See Section “Defer” in *prop*
 SetDiff : See Section “PropLib” in *prop*
 SetGet : See Section “PropLib” in *prop*
 SetKeyOfEnv : See Section “Queries” in *name*
 SetKind : See Section “Kind” in *prop*
 SetOnce : See Section “PropLib” in *prop*
 ShieldClause : See Section “CONSTITUENT(S)” in *lidoref*
 ShowBinding : See Section “Module for Testing Name Analysis” in *news4.1*
 SimpExpr : See Section “Simple Expressions” in *lidoref*
 SimpExpr : See Section “Terminal Access” in *lidoref*
 Simple Expressions : See Section “Simple Expressions” in *lidoref*
 Single : See Section “List” in *adt*
 SingleFctName : See Section “CONSTITUENT(S)” in *lidoref*
 SingleList : See Section “Adt” in *news4.3*
 SingleTYPEList : See Section “ModLib” in *news4.2*
 Solutions of Common Problems : See Section “top” in *problems*
 Sort : See Section “Sort added” in *news4.1*
 SortTYPE : See Section “Sort” in *problems*
 Source Language Structure : See Section “Structure” in *ptg*

Specification : See Section “Overall Structure” in *lidoref*
 Specification modules — Expression : See Section “Expressions” in *type*
 Specification modules — Expression : See Section “IdentifyOperator” in *type*
 Specification modules — StructEquiv : See Section “TypeEq” in *type*
 Specification modules — Typing : See Section “BasicType” in *type*
 Specifications : See Section “Specifications” in *lex*
 SrcBufPtr : See Section “source” in *lib*
 SrcBuffer : See Section “source” in *lib*
 Stack : See Section “ModLib3_6” in *oldmod*
 StackArray : See Section “Stack” in *adt*
 StackElement : See Section “Stack” in *adt*
 StackEmpty : See Section “Stack” in *adt*
 StackPop : See Section “Stack” in *adt*
 StackPush : See Section “Stack” in *adt*
 StackSize : See Section “Stack” in *adt*
 StackTop : See Section “Stack” in *adt*
 Status Display : See Section “The Main Window” in *gorto*
 Storage.h : See Section “datamap” in *lib*
 StorageAlignment : See Section “datamap” in *lib*
 StorageRequired : See Section “datamap” in *lib*
 StorageSize : See Section “datamap” in *lib*
 StrArith : See Section “StrArith” in *news4.4*
 Strategy Menus : See Section “The Main Window” in *gorto*
 String Buffer : See Section “StringOut” in *output*
 String Literals : See Section “PtgCommon Examples” in *output*
 String command : See Section “String Table” in *mon*
 StringTable : See Section “String value options” in *clp*
 StringTable : See Section “storage” in *lib*
 StringTable : See Section “Expressions” in *ptg*
 Strings : See Section “Strings” in *problems*
 Strings command : See Section “String Table” in *mon*
 Structural equivalence : See Section “UserTypes” in *type*
 Sum : See Section “List” in *adt*
 SumFct : See Section “List” in *adt*
 SymbKind : See Section “Symbol Specifications” in *lidoref*
 SymbName : See Section “Productions” in *lidoref*
 SymbName : See Section “Symbol Specifications” in *lidoref*
 SymbNameList : See Section “CONSTITUENT(S)” in *lidoref*
 Symbol Mapping : See Section “Symbol Mapping” in *syntax*
 Symbol Menu : See Section “Symbols” in *gorto*
 Symbol Specifications : See Section “Symbol Specifications” in *lidoref*
 SymbolRef : See Section “Attributes” in *lidoref*
 Symbols : See Section “Computed Subtrees” in *lidoref*
 Symbols : See Section “Productions” in *lidoref*
 Syntax : See Section “Syntax” in *lidoref*
 Syntax Errors : See Section “Options” in *cola*
 Syntax of PTG Specifications : See Section “Syntax” in *ptg*

TAIL : See Section “Chain” in *comptrees*
 TCL : See Section “Breakpoints” in *mon*
 TERM : See Section “Tree Construction” in *comptrees*
 TERM : See Section “Value” in *comptrees*
 TEXTSTART : See Section “source” in *lib*
 THIS : See Section “Basic” in *comptrees*
 TPNUL : See Section “Predefined” in *tp*
 TPNode : See Section “Predefined” in *tp*
 TPNull : See Section “Predefined” in *tp*
 TREEBUILD : See Section “Refactor” in *news4.8*
 Table : See Section “Hashing” in *news4.4*
 Tail : See Section “List” in *adt*
 Tcl specifications : See Section “Monitoring user-defined types” in *news4.3*
 TermFct : See Section “Predefined Entities” in *lidoref*
 Terminal Access : See Section “Terminal Access” in *lidoref*
 Terminator : See Section “Computations” in *lidoref*
 Texinfo : See Section “Documentation” in *pp*
 Token command : See Section “Lexical Structure” in *mon*
 Token names in Noosa : See Section “Main window command changes” in *news4.3*
 TotalCnt : See Section “OccCnt” in *prop*
 Trace of Dependencies : See Section “Symbols” in *gorto*
 Tracing Dependencies : See Section “Productions” in *gorto*
 Trans : See Section “PropLib” in *prop*
 TransDefer : See Section “Defer” in *prop*
 TransferContext : See Section “ExpressionSymbol” in *type*
 TransferContext : See Section “Expressions” in *type*
 TransferContext : See Section “OpndExprListRoot” in *type*
 Tree Construction Functions : See Section “Tree Construction Functions” in *lidoref*
 Trees menu : See Section “Trees” in *mon*
 Type : See Section “ModLib3_8” in *oldmod*
 Type : See Section “FuncType” in *type*
 Type : See Section “OpDef” in *type*
 Type : See Section “Support” in *type*
 Type : See Section “TypeIdDepend” in *type*
 Type : See Section “TypeUse” in *type*
 Type equivalence, name : See Section “UserTypes” in *type*
 Type equivalence, structural : See Section “UserTypes” in *type*
 TypeDefDefId : See Section “UserTypes” in *type*
 TypeDefUseId : See Section “TypeUse” in *type*
 TypeDefUseId : See Section “UserTypes” in *type*
 TypeDenotation : See Section “FuncType” in *type*
 TypeDenotation : See Section “OilClass” in *type*
 TypeDenotation : See Section “OpDef” in *type*
 TypeDenotation : See Section “TypeDenot” in *type*
 TypeDenotation : See Section “UserTypes” in *type*
 TypeIsSet : See Section “TypedIdDepend” in *type*
 TypeName : See Section “CONSTITUENT(S)” in *lidoref*

TypeName : See Section “Types and Classes of Attributes” in *lidoref*
 Typed Entities : See Section “BasicType” in *type*
 Typed Insertion Points : See Section “Typed” in *ptg*
 TypedDefId : See Section “BasicType” in *type*
 TypedDefId : See Section “TypedDefId” in *type*
 TypedDefinition : See Section “BasicType” in *type*
 TypedDefinition : See Section “TypedDefId” in *type*
 TypedDefinition : See Section “TypedEntityInfo” in *type*
 TypedIdDef : See Section “TypedEntityInfo” in *type*
 TypedUseId : See Section “BasicType” in *type*
 TypedUseId : See Section “ExpressionSymbol” in *type*
 TypedUseId : See Section “TypeIdDepend” in *type*
 TypedUseId : See Section “TypedEntityInfo” in *type*
 TypedUseId : See Section “TypedUseId” in *type*
 Types and Classes of Attributes : See Section “Types and Classes of Attributes” in *lidoref*
 Unique : See Section “Unique” in *deftbl*
 Unique : See Section “UniqueCode” in *deftbl*
 Unique : See Section “Instantiation” in *novice*
 Unique : See Section “ModLib3_6” in *oldmod*
 Unique : See Section “Unique” in *prop*
 Unix newline : See Section “Inserting Arbitrary” in *fw*
 Using LIDO CHAINS : See Section “CHAIN” in *ptg*
 Using LIDO CONSTITUENTS : See Section “CONSTITUENTS” in *ptg*
 VALUE : See Section “Operations” in *deftbl*
 VOID : See Section “State” in *comptrees*
 VOID attribute not allowed here : See Section “Void Attributes” in *comptrees*
 VReset : See Section “PropLib” in *prop*
 VerifyLevel : See Section “Editing files” in *mon*
 Visit-sequence Menu : See Section “Visit-sequences” in *gorto*
 VoidPtr : See Section “List” in *adt*
 WARNING : See Section “error” in *lib*
 WARNING error severity : See Section “Refactor” in *news4.8*
 WINEDITOR : See Section “Options” in *info*
 WITH clause : See Section “Constituents” in *comptrees*
 WITH functions : See Section “CONSTITUENTS” in *ptg*
 Windows : See Section “Windows” in *news4.4*
 WithClause : See Section “CONSTITUENT(S)” in *lidoref*
 X resources : See Section “X resources” in *mon*
 X resources for Noosa : See Section “Configuring Noosa” in *news4.3*
 X-Windows : See Section “(gorto)Using GORTO within ELI” in *X11*
 XML : See Section “Unparsers” in *news4.4*
 \040+ : See Section “Definitions of Descriptions” in *lex*
 _obstack : See Section “obstack” in *lib*
 “unavailable” error report : See Section “test” in *pp*
 abort : See Section “Diagnostics” in *pp*
 abstract data type : See Section “Interrelationship” in *oil*
 abstract data type : See Section “Actions” in *syntax*

abstract syntax tree display : See Section “Monitoring” in *news4.1*
abstracting from literals : See Section “Abstraction” in *syntax*
abstree : See Section “abstree” in *pp*
absyntax : See Section “absyntax” in *pp*
accepting a basic symbol : See Section “Parsing” in *syntax*
accumulating attribute : See Section “Accu” in *comptrees*
accumulating attribute : See Section “Accumulating Computations” in *lidoref*
accumulating computations : See Section “Accu” in *comptrees*
accumulating computations : See Section “Accumulating Computations” in *lidoref*
accumulating computations : See Section “Computations” in *lidoref*
action : See Section “Actions” in *tp*
action : See Section “Rules” in *tp*
action macro : See Section “Implementation” in *tp*
action routine : See Section “Implementation” in *tp*
action signature : See Section “Evaluation” in *tp*
action signature : See Section “Implementation” in *tp*
actions during parsing : See Section “Actions” in *syntax*
additional information : See Section “Use” in *idem*
adjacent context : See Section “Tree” in *comptrees*
alignment : See Section “obstack” in *lib*
alterations : See Section “ChangingFiles” in *faq*
altering files : See Section “help” in *pp*
alternation : See Section “Bar” in *lex*
alternation : See Section “Slash” in *syntax*
ambiguity : See Section “Conflicts” in *syntax*
ambiguity resolved by grammar changes : See Section “Changes” in *syntax*
ambiguity resolved by modifications : See Section “Modifiers” in *syntax*
anonymous entities : See Section “Entities” in *deftbl*
application language, selecting : See Section “DeriveTree” in *idem*
application languages, CPP : See Section “treeIng” in *idem*
application languages, Java : See Section “treeIng” in *idem*
application languages, XML : See Section “treeIng” in *idem*
application languages, other : See Section “treeIng” in *idem*
applied occurrences : See Section “Preconditions” in *name*
arbitrary actions during parsing : See Section “Actions” in *syntax*
arbitrary characters, inserting into text : See Section “Inserting Arbitrary” in *fw*
arg : See Section “arg” in *pp*
argc : See Section “Command line interface” in *clp*
argument separator : See Section “Combine” in *idem*
argument separator : See Section “Compute” in *idem*
argument separator : See Section “Use” in *idem*
argument type : See Section “Possible Types” in *oil*
argument type : See Section “Signature” in *oil*
arguments : See Section “Controlling Your Program” in *mon*
argv : See Section “Command line interface” in *clp*
arithmetic expressions : See Section “Structure” in *tp*
arithmetic on character strings : See Section “StrArith” in *problems*

arity : See Section “Structure” in *tp*
 aspect : See Section “Aspects” in *mon*
 association rules : See Section “Association” in *syntax*
 attribute : See Section “Depend” in *comptrees*
 attribute : See Section “Attribution” in *novice*
 attribute Cnt : See Section “OccCnt” in *prop*
 attribute CountResult : See Section “Counter” in *problems*
 attribute DefCond : See Section “CScope” in *name*
 attribute Elem : See Section “LidoList” in *adt*
 attribute Env : See Section “AlgScope” in *name*
 attribute Env : See Section “CScope” in *name*
 attribute GENTREE : See Section “Computed Subtrees” in *lidoref*
 attribute GotCnt : See Section “OccCnt” in *prop*
 attribute GotKeys : See Section “AlgScope” in *name*
 attribute GotKeys : See Section “CScope” in *name*
 attribute GotKind : See Section “Kind” in *prop*
 attribute GotKind : See Section “KindSet” in *prop*
 attribute GotLocKeys : See Section “AlgScope” in *name*
 attribute HasKindSet : See Section “KindSet” in *prop*
 attribute HasKind : See Section “Kind” in *prop*
 attribute IncrementCnt : See Section “ObjCnt” in *prop*
 attribute Increment : See Section “Counter” in *problems*
 attribute InitCount : See Section “Counter” in *problems*
 attribute InitObjCnt : See Section “ObjCnt” in *prop*
 attribute IsFirstOcc : See Section “FirstOcc” in *prop*
 attribute Key : See Section “Instantiation” in *modlib*
 attribute Key : See Section “AlgScope” in *name*
 attribute Key : See Section “Basic Scope Rules” in *name*
 attribute Key : See Section “CScope” in *name*
 attribute Key : See Section “Usage” in *prop*
 attribute KindSet : See Section “KindSet” in *prop*
 attribute Kind : See Section “Kind” in *prop*
 attribute Kind : See Section “KindSet” in *prop*
 attribute ListTail : See Section “LidoList” in *adt*
 attribute List : See Section “LidoList” in *adt*
 attribute ObjNo : See Section “ObjCnt” in *prop*
 attribute ScopeKey : See Section “ScopeProp” in *name*
 attribute Scope : See Section “AlgScope” in *name*
 attribute Scope : See Section “CScope” in *name*
 attribute Scope : See Section “ScopeProp” in *name*
 attribute Sym : See Section “AlgScope” in *name*
 attribute Sym : See Section “CScope” in *name*
 attribute Sym : See Section “Preconditions” in *name*
 attribute Sym : See Section “LeafPtg” in *output*
 attribute TakeIt : See Section “LidoList” in *adt*
 attribute TotalCnt : See Section “OccCnt” in *prop*
 attribute TotalObjNo : See Section “ObjCnt” in *prop*

attribute Unique : See Section “Unique” in *prop*
 attribute Bind : See Section “AlgScope” in *name*
 attribute Bind : See Section “Basic Scope Rules” in *name*
 attribute Bind : See Section “CScope” in *name*
 attribute Count : See Section “Counter” in *problems*
 attribute DeferredKey : See Section “Defer” in *prop*
 attribute GotDeferProp : See Section “Defer” in *prop*
 attribute GotInhScopes : See Section “CInh” in *name*
 attribute GotScopeProp : See Section “CScopeProp” in *name*
 attribute GotScopeProp : See Section “ScopeProp” in *name*
 attribute GotVisibleKeys : See Section “AlgInh” in *name*
 attribute GotVisibleKeysNest : See Section “AlgInh” in *name*
 attribute InhPrecond : See Section “BuInh” in *name*
 attribute InheritOk : See Section “Inheritance of Scopes” in *name*
 attribute InnerScope : See Section “Inheritance of Scopes” in *name*
 attribute NewScope : See Section “BuInh” in *name*
 attribute OpenPrecond : See Section “BuScopeProp” in *name*
 attribute OuterScope : See Section “Inheritance of Scopes” in *name*
 attribute ScopeKey : See Section “BuScopeProp” in *name*
 attribute ScopeKey : See Section “Inheritance of Scopes” in *name*
 attribute class : See Section “Value” in *comptrees*
 attribute class : See Section “Attributes” in *lidoref*
 attribute class : See Section “Symbol Specifications” in *lidoref*
 attribute class : See Section “Types and Classes of Attributes” in *lidoref*
 attribute class in conflict : See Section “Attribute Class Conflict” in *comptrees*
 attribute computations : See Section “Attribute Computations and Plain Computations” in *lidoref*
 attribute computations : See Section “Computations” in *lidoref*
 attribute grammar : See Section “Instantiation” in *novice*
 attribute grammar : See Section “Output” in *novice*
 attribute type : See Section “Value” in *comptrees*
 attribute type : See Section “Attribute Computations and Plain Computations” in *lidoref*
 attribute type : See Section “CONSTITUENT(S)” in *lidoref*
 attribute type : See Section “INCLUDING” in *lidoref*
 attribute type : See Section “Types and Classes of Attributes” in *lidoref*
 attribute types : See Section “Evaluation” in *tp*
 attribute value display : See Section “Monitoring” in *news4.1*
 attribute values : See Section “Trees” in *mon*
 attribute, IdemOrigPtg : See Section “Compute” in *idem*
 attributes : See Section “idemsep” in *idem*
 attributes : See Section “treedel” in *idem*
 attributes : See Section “treesep” in *idem*
 attributes : See Section “Attributes” in *lidoref*
 attributes : See Section “Attributes” in *tp*
 attributes : See Section “Rules” in *tp*
 attributes : See Section “Tree” in *tp*
 autoloading handlers : See Section “Breakpoints” in *mon*

autoloading handlers : See Section “User Initialisation” in *mon*
auxM2StringDQ : See Section “Definitions of Descriptions” in *lex*
auxM2StringSQ : See Section “Definitions of Descriptions” in *lex*
auxiliary scanner : See Section “Auxiliary Scanners” in *lex*
axiom : See Section “Syntactic analysis” in *novice*
axiom : See Section “Phrases” in *syntax*
axiom, recognition of : See Section “Parsing” in *syntax*
b : See Section “Help-C-l” in *info*
backslash : See Section “Quoting” in *lex*
balancing : See Section “Possible Types” in *oil*
basic scope rules : See Section “Basic Scope Rules” in *name*
basic symbol : See Section “printtokens” in *pp*
basic symbol : See *syntax*
basic symbol, how parser accepts : See Section “Parsing” in *syntax*
basis : See Section “Derivation” in *idem*
binding : See Section “Environment Module” in *name*
binding : See Section “Environment Module Enhancements” in *news4.1*
block print module : See Section “treesep” in *idem*
block-structure : See Section “PrettyPrint” in *output*
boolean : See Section “Boolean options” in *clp*
boolean option : See Section “Boolean options” in *clp*
boolean options repeated : See Section “Repeated options” in *clp*
booleans : See Section “Boolean options” in *clp*
bottom-up : See Section “Include” in *input*
bottom-up : See Section “top” in *input*
bottom-up : See Section “Basic Scope Rules” in *name*
bottom-up : See Section “BuInh” in *name*
bottom-up : See Section “BuScope” in *name*
bottom-up : See Section “BuScopeProp” in *name*
bottom-up : See Section “Bottom-up Evaluation” in *news4.1*
bottom-up computations : See Section “Attribute Computations and Plain Computations” in *lidoref*
bottom-up computations : See Section “Computed Subtrees” in *lidoref*
breakpoints : See Section “Breakpoints” in *mon*
browser session : See Section “parex” in *novice*
browsing attribute values : See Section “Trees” in *mon*
browsing attribute values : See Section “Monitoring” in *news4.1*
browsing chain values : See Section “Trees” in *mon*
browsing non-standard types : See Section “Browsing non-standard types” in *mon*
browsing the abstract tree : See Section “Trees” in *mon*
built-in symbols : See Section “Canned Descriptions” in *lex*
c : See Section “Complaining” in *info*
cache : See Section “Strategies” in *novice*
cache : See *sysadmin*
cache name : See Section “CacheName” in *sysadmin*
cache packages : See Section “Capability” in *sysadmin*
cache size : See Section “CacheSize” in *faq*

canned description : See Section “GLA specification” in *novice*
canned symbols : See Section “Canned Descriptions” in *lex*
case-insensitivity : See Section “fold” in *pp*
cc : See Section “cc” in *pp*
cc.flags : See Section “cc_flags” in *pp*
chain : See Section “CHAIN” in *lidoref*
chain attributes : See Section “Trees” in *mon*
chain production : See Section “Bottom-up Evaluation” in *news4.1*
chain productions : See Section “Tree Construction Functions” in *lidoref*
chain rule : See Section “Chain” in *tp*
chain rule, omitted : See Section “Compute” in *idem*
chain rules : See Section “Chain Rules” in *syntax*
changing a component : See Section “Monitoring Support” in *mon*
char : See Section “Typed” in *ptg*
character sequences : See Section “Lexical analysis” in *novice*
character sequences : See Section “Specifications” in *novice*
character string arithmetic : See Section “StrArith” in *problems*
character strings : See Section “kwd” in *pp*
child : See Section “Attribution” in *novice*
child, omitting in the unparsed output : See Section “Combine” in *idem*
chunk size : See Section “obstack” in *lib*
class : See Section “Value” in *comptrees*
class definition : See Section “CLASS” in *oil*
class definition : See Section “Statements” in *oil*
class of attributes : See Section “Attributes” in *lidoref*
class of attributes : See Section “Symbol Specifications” in *lidoref*
class of attributes : See Section “Types and Classes of Attributes” in *lidoref*
class symbol computation : See Section “Compute” in *idem*
classes : See Section “Classes” in *lex*
classes : See Section “Language” in *oil*
client, Odin : See Section “Execution” in *sysadmin*
cloning keys : See Section “Keys” in *deftbl*
clp : See Section “Command” in *news4.8*
cmd : See Section “cmd” in *pp*
coercion : See Section “COERCION” in *oil*
coercion : See Section “Example” in *oil*
coercion : See Section “Possible Types” in *oil*
coercion : See Section “Simple Example” in *oil*
coercion operator : See Section “Possible Types” in *oil*
coercion operator definition : See Section “COERCION” in *oil*
coercion operator definition : See Section “Statements” in *oil*
coercion sequence : See Section “Coercion Sequences” in *oil*
coercion sequence : See Section “Library Types” in *oil*
coercion sequence : See Section “Simple Example” in *oil*
coercion sequence, empty : See Section “Coercion Sequences” in *oil*
coercion sequence, error : See Section “Coercion Sequences” in *oil*
coercion sequence, head of : See Section “Coercion Sequences” in *oil*

coercion sequence, tail of : See Section “Coercion Sequences” in *oil*
coercions : See Section “Chain Rules” in *syntax*
collections, of packages : See Section “Collections” in *sysadmin*
colours : See Section “X resources” in *mon*
combining properties : See Section “Data” in *deftbl*
combining related specifications : See Section “Generation” in *novice*
comma expression : See Section “Structure” in *tp*
command line : See Section “Command line interface” in *clp*
command line : See Section “Refactor” in *news4.8*
command line argument : See Section “Specifications” in *novice*
command line arguments : See Section “CmdLineIncl” in *input*
command line arguments : See Section “Cache” in *novice*
command line format : See Section “Command line format” in *clp*
command line parameters : See Section “arg” in *pp*
command line processor : See Section “Command” in *news4.8*
command-line options : See Section “Controlling Your Program” in *mon*
command-line processing : See Section “Clp” in *news4.1*
commands, *eli* : See Section “EliCmd” in *news4.6*
commands, *eli* : See Section “CacheName” in *sysadmin*
commands, *odin* : See Section “Odin” in *sysadmin*
commands, *uname* : See Section “CacheName” in *sysadmin*
comment : See Section “Atoms” in *novice*
comments : See Section “Definition” in *deftbl*
comments : See Section “Comments” in *fw*
comments : See Section “Syntax” in *lidoref*
comments : See Section “Patterns” in *ptg*
comments : See Section “Syntax” in *ptg*
common file system : See Section “CacheName” in *sysadmin*
commutativity : See Section “Commutativity” in *tp*
commutativity : See Section “Rules” in *tp*
compatibility with token processor : See Section “Combine” in *idem*
compiler messages : See Section “C-Errors” in *comptrees*
complaining : See Section “Complaining” in *info*
complement : See Section “Classes” in *lex*
completion : See Section “Completion” in *info*
computation : See Section “Depend” in *comptrees*
computation : See Section “Attribution” in *novice*
computation : See Section “Specifications” in *novice*
computation, *IdemPtg* : See Section “Changing” in *idem*
computation, *IdemPtg* : See Section “Compute” in *idem*
computation, class symbol : See Section “Compute” in *idem*
computation, rule : See Section “Compute” in *idem*
computation, tree symbol : See Section “Compute” in *idem*
computational role : See Section “Type” in *news4.5*
computations for LISTOF productions : See Section “idemdel” in *idem*
computations for LISTOF productions : See Section “treedel” in *idem*
computations for plain productions : See Section “idemsep” in *idem*

computations for plain productions : See Section “treesep” in *idem*
computed sub-tree : See Section “Multiple” in *idem*
computed subtrees : See Section “nomain” in *news4.4*
concatenation : See Section “Several” in *lex*
concrete grammar : See Section “Tree Construction” in *comptrees*
concrete grammar : See Section “Rule Specifications” in *lidoref*
conditional compilation : See Section “Specifications” in *novice*
conditional expression : See Section “Fragment” in *tp*
conditional expression : See Section “Structure” in *tp*
conflict : See Section “Conflicts” in *syntax*
conflict due to user-defined actions : See Section “Actions” in *syntax*
conflict solved by \$ modification : See Section “Dollar” in *syntax*
conflict solved by @ modification : See Section “At” in *syntax*
conflict, shift-reduce : See Section “parex” in *novice*
connections : See Section “Options” in *cola*
consistent renaming : See Section “top” in *name*
constructing a separator module : See Section “idemsep” in *idem*
constructing generic modules : See Section “inst” in *pp*
construction function : See Section “Construction” in *tp*
consyntax : See Section “consyntax” in *pp*
context, lower : See Section “Instantiation” in *novice*
context-free grammar : See Section “Structure” in *novice*
context-free grammar : See Section “Syntactic analysis” in *novice*
context-free grammar : See Section “Phrases” in *syntax*
control characters, inserting into text : See Section “Inserting Arbitrary” in *fw*
controlled execution : See Section “Debugging” in *pp*
controlling execution : See Section “Controlling Your Program” in *mon*
controlling program : See Section “Controlling Your Program” in *mon*
copy commands : See Section “copy” in *ui*
core : See Section “Debugging” in *pp*
cost : See Section “Patterns” in *tp*
cost : See Section “Rules” in *tp*
count occurrences of objects : See Section “OccCnt” in *prop*
coupling between specifications : See Section “Generation” in *novice*
creating keys : See Section “Keys” in *deftbl*
creating new names : See Section “Name Functions” in *oil*
critical attributes : See Section “Symbols” in *gorto*
csm.h : See Section “storage” in *lib*
cumulative column : See Section “Position” in *lex*
cumulative coordinates : See Section “Input Text Display” in *mon*
curpos : See Section “error” in *lib*
customisation : See Section “User Initialisation” in *mon*
cyclic dependencies : See Section “Cycles” in *comptrees*
cyclic dependencies : See Section “Attribute Computations and Plain Computations” in *lidoref*
cyclic dependencies : See Section “Iterations” in *lidoref*
d : See Section “Help-Adv” in *info*

daVinci : See [Section “Unparsers”](#) in *news4.4*
dangling else problem : See [Section “Shift-reduce”](#) in *syntax*
dangling else solved by \$ modification : See [Section “Dollar”](#) in *syntax*
dangling else solved by grammar change : See [Section “Changes”](#) in *syntax*
dapto scoping rules : See [Section “Dapto Grammar”](#) in *mon*
dash : See [Section “Classes”](#) in *lex*
database : See [Section “Database”](#) in *mon*
database : See [Section “Implementing Interfaces”](#) in *mon*
dbx : See [Section “Debugging”](#) in *pp*
dbxtool : See [Section “Debugging”](#) in *pp*
debug : See [Section “debug”](#) in *pp*
debug : See [Section “monitor”](#) in *pp*
debugging : See [Section “Help”](#) in *novice*
debugging : See [Section “Products”](#) in *novice*
debugging : See [Section “debugex”](#) in *novice*
debugging : See [Section “Debugging”](#) in *pp*
debugging interprocess communication : See [Section “Implement”](#) in *sysadmin*
declaration : See [Section “Declarations”](#) in *tp*
declarations : See [Section “CONSTITUENTS”](#) in *ptg*
declarations, scope issues : See [Section “Scope”](#) in *syntax*
decomposition : See [Section “Subproblems”](#) in *novice*
default behavior for white space : See [Section “White Space”](#) in *lex*
default interface : See [Section “Default behaviour”](#) in *clp*
default special character : See [Section “Special”](#) in *fw*
define : See [Section “define”](#) in *pp*
defining occurrences : See [Section “Preconditions”](#) in *name*
definition before application : See [Section “SetFirst”](#) in *prop*
definition table : See [Section “Property storage”](#) in *novice*
definition table : See [Section “Specifications”](#) in *novice*
definition table design : See [Section “Criteria”](#) in *deftbl*
deftbl.h : See [Section “Module”](#) in *deftbl*
dependence among products : See [Section “Derivation”](#) in *pp*
dependencies : See [Section “Attribute Computations and Plain Computations”](#) in *lidoref*
dependency : See [Section “Depend”](#) in *comptrees*
dependent computations : See [Section “Depend”](#) in *comptrees*
derivation : See [Section “Phrases”](#) in *syntax*
derivation, structural unparser : See [Section “DeriveTree”](#) in *idem*
derivation, structure definition : See [Section “DeriveDefn”](#) in *idem*
derivation, textual unparser : See [Section “DeriveIdem”](#) in *idem*
derivation, unparser : See [Section “Derivation”](#) in *idem*
derived file object, output : See [Section “Interactive”](#) in *novice*
derived list object, output : See [Section “Interactive”](#) in *novice*
derived object : See [Section “Products”](#) in *novice*
derived object : See [Section “objects”](#) in *ui*
derived object cache : See [Section “Strategies”](#) in *novice*
descriptions of subproblems : See [Section “Specifications”](#) in *novice*
development process : See [Section “Example”](#) in *novice*

dfltrepar.c : See Section “Error Recovery” in *syntax*
diagnostic products : See Section “Diagnostics” in *pp*
directives : See Section “ignore” in *pp*
directories : See Section “CmdLineIncl” in *input*
documentation : See Section “Documentation” in *clp*
dofold : See Section “identifier” in *lib*
domain-specific modules : See Section “inst” in *pp*
dot : See Section “Classes” in *lex*
double : See Section “Complex Example” in *oil*
double : See Section “Typed” in *ptg*
double quote : See Section “Quoting” in *lex*
dumpidn : See Section “identifier” in *lib*
dumpstr : See Section “storage” in *lib*
dvi : See Section “dvi” in *pp*
e : See Section “Examples” in *info*
e : See Section “Expert” in *info*
edit command : See Section “help” in *pp*
editing : See Section “parex” in *novice*
editing a file object : See Section “Interactive” in *novice*
editing files : See Section “Editing files” in *mon*
editing files while monitoring : See Section “Monitoring” in *news4.1*
empty .clp files : See Section “Clp” in *news4.1*
empty coercion sequence : See Section “Coercion Sequences” in *oil*
empty input text, representation of : See Section “At” in *syntax*
empty output : See Section “PTG” in *news4.0*
empty rules : See Section “CHAIN in Empty Rules” in *news4.1*
empty specification : See Section “Command line format” in *clp*
encodings of non-literals : See Section “Building Processors” in *lex*
enter event : See Section “Time Profiles” in *mon*
entity : See Section “Entities” in *novice*
entity : See Section “Property storage” in *novice*
environment module : See Section “Keys” in *deftbl*
environment variable *ODIN_LOCALIPC* : See Section “Implement” in *sysadmin*
environment variable, *ODINPATH* : See Section “Capability” in *sysadmin*
environment variable, *ODINPATH* : See Section “Odin” in *sysadmin*
environment variable, *ODINVIEW* : See Section “CacheName” in *sysadmin*
environment variable, *ODIN* : See Section “CacheName” in *sysadmin*
envmod : See Section “Environment Module” in *name*
envmod : See Section “Environment Module Enhancements” in *news4.1*
envmod.h : See Section “environment” in *lib*
err : See Section “Products” in *novice*
err.h : See Section “error” in *lib*
error : See Section “error” in *pp*
error coercion : See Section “Coercion Sequences” in *oil*
error coercion sequence : See Section “Coercion Sequences” in *oil*
error format : See Section “Usage options” in *clp*
error messages : See Section “Errors” in *comptrees*

error messages : See [Section “Messages”](#) in *mon*
error messages : See [Section “Message”](#) in *problems*
error report : See [Section “error”](#) in *pp*
error report : See [Section “warning”](#) in *pp*
error reported in a modification : See [Section “Modifiers”](#) in *syntax*
error reports during parsing : See [Section “Phrases”](#) in *syntax*
error severity : See [Section “Instantiation”](#) in *novice*
error severity : See [Section “Output”](#) in *novice*
error severity level WARNING : See [Section “Refactor”](#) in *news4.8*
event : See [Section “Breakpoints”](#) in *mon*
event : See [Section “Event Types”](#) in *mon*
event counting : See [Section “Frequency Profiles”](#) in *mon*
event handlers : See [Section “Breakpoints”](#) in *mon*
event parameters : See [Section “Breakpoints”](#) in *mon*
event type : See [Section “Event Types”](#) in *mon*
exact right context : See [Section “Modifiers”](#) in *syntax*
examining attributes : See [Section “Trees”](#) in *mon*
example : See [Section “Examples”](#) in *info*
example : See [Section “Example”](#) in *ptg*
example application : See [Section “Example”](#) in *deftbl*
example language : See [Section “Structure”](#) in *ptg*
example of a reduce-reduce conflict : See [Section “Reduce-reduce”](#) in *syntax*
example of a shift-reduce conflict : See [Section “Shift-reduce”](#) in *syntax*
example of a type-‘con’ file : See [Section “Con”](#) in *syntax*
example of a type-‘map’ file : See [Section “Sym”](#) in *syntax*
example of debugging : See [Section “debugex”](#) in *novice*
example of editing : See [Section “parex”](#) in *novice*
example of obtaining help : See [Section “parex”](#) in *novice*
example of requesting source code : See [Section “symbex”](#) in *novice*
example of user-defined actions : See [Section “Actions”](#) in *syntax*
example operation declaration : See [Section “GetCode”](#) in *deftbl*
examples : See [Section “Example”](#) in *modlib*
examples : See [Section “top”](#) in *modlib*
examples : See [Section “Basic Scope Rules”](#) in *name*
examples : See [Section “top”](#) in *name*
exe : See [Section “SO”](#) in *news4.8*
exe : See [Section “exe”](#) in *pp*
executable program : See [Section “exe”](#) in *pp*
executable program : See [Section “so”](#) in *pp*
executable target : See [Section “Session”](#) in *novice*
executable target : See [Section “Odinfile”](#) in *ui*
execution monitoring : See [Section “Monitoring”](#) in *pp*
execution, of Odin : See [Section “Execution”](#) in *sysadmin*
exiting Noosa : See [Section “Invoking Noosa”](#) in *mon*
expected argument type : See [Section “Signature”](#) in *oil*
expert system : See [Section “Derivation”](#) in *pp*
explanation of reports : See [Section “help”](#) in *pp*

exported identifiers : See Section “Predefined” in *ptg*
exported identifiers : See Section “Predefined” in *tp*
expressions : See Section “Expressions” in *lidoref*
expressions for derived objects : See Section “objects” in *ui*
external modifications, informing Eli : See Section “test” in *pp*
f : See Section “Help-Adv” in *info*
feInfo : See Section “LIGA” in *news4.5*
feedback : See Section “Derivation” in *pp*
feedback to the user : See Section “Session” in *novice*
file ‘ENV’ : See Section “Implement” in *sysadmin*
file ‘PKGLST’ : See Section “Capability” in *sysadmin*
file ‘SOCKET’ : See Section “Communicate” in *sysadmin*
file ‘SOCKET’ : See Section “Implement” in *sysadmin*
file format : See Section “Dapto Grammar” in *mon*
file formats : See Section “Grammars” in *syntax*
file name : See Section “Frame” in *ptg*
file object, editing : See Section “Interactive” in *novice*
file object, output : See Section “Interactive” in *novice*
file target : See Section “Session” in *novice*
file target : See Section “Odinfile” in *ui*
file types: .finl : See Section “Refactor” in *news4.8*
file types: .init : See Section “Refactor” in *news4.8*
file, ‘PKGLST’ : See Section “Collections” in *sysadmin*
filename : See Section “Output Files” in *fw*
files, non-product : See Section “Output Files” in *fw*
files, non-product : See Section “FunnelWeb” in *news4.1*
finalisation : See Section “Breakpoints” in *mon*
finalization : See Section “Specifications” in *novice*
finit : See Section “Breakpoints” in *mon*
finite-state machine : See Section “Parsing” in *syntax*
finlBuf : See Section “source” in *lib*
flat range : See Section “Basic Scope Rules” in *name*
float : See Section “Typed” in *ptg*
floating point numbers : See Section “Data” in *ptg*
fold : See Section “fold” in *pp*
fonts : See Section “X resources” in *mon*
fonts in Noosa : See Section “Monitoring” in *news4.1*
foreign parser : See Section “Foreign” in *news4.8*
formal parameter lists : See Section “Parameter Lists” in *fw*
formatted file : See Section “dvi” in *pp*
formatted file : See Section “ps” in *pp*
formatting strategy : See Section “Use” in *idem*
frame : See Section “top” in *lib*
free : See Section “obstack” in *lib*
frequency profile : See Section “Frequency Profiles” in *mon*
function : See Section “Value” in *comptrees*
function AddE1IS : See Section “IntSet” in *adt*

function AddElemToBitSet : See Section “BitSet” in *adt*
function AddRangeToBitSet : See Section “BitSet” in *adt*
function ApplyToBitSet : See Section “BitSet” in *adt*
function C_outchar : See Section “OutStr” in *output*
function C_outstr : See Section “OutStr” in *output*
function CardIS : See Section “IntSet” in *adt*
function CardOfBitSet : See Section “BitSet” in *adt*
function ComplBitSet : See Section “BitSet” in *adt*
function ComplToBitSet : See Section “BitSet” in *adt*
function ConsIS : See Section “IntSet” in *adt*
function DefInTable : See Section “Table” in *adt*
function DisjIS : See Section “IntSet” in *adt*
function DynAlloc : See Section “DynSpace” in *adt*
function DynClear : See Section “DynSpace” in *adt*
function ElemInBitSet : See Section “BitSet” in *adt*
function ElemToBitSet : See Section “BitSet” in *adt*
function EmptyBitSet : See Section “BitSet” in *adt*
function EmptyIS : See Section “IntSet” in *adt*
function EmptyIntersectBitSet : See Section “BitSet” in *adt*
function EqualBitSet : See Section “BitSet” in *adt*
function EqualIS : See Section “IntSet” in *adt*
function FindFile : See Section “CmdLineIncl” in *input*
function FinlMap : See Section “Map” in *adt*
function FreeBitSet : See Section “BitSet” in *adt*
function FreeMemBitSet : See Section “BitSet” in *adt*
function GetCurrTok : See Section “CurrTok” in *input*
function GetMap : See Section “Map” in *adt*
function InIS : See Section “IntSet” in *adt*
function InclIS : See Section “IntSet” in *adt*
function InitDynSpace : See Section “DynSpace” in *adt*
function InitMapValues : See Section “Map” in *adt*
function InitMap : See Section “Map” in *adt*
function InitTable : See Section “Table” in *adt*
function InterIS : See Section “IntSet” in *adt*
function IntersectBitSet : See Section “BitSet” in *adt*
function IntersectToBitSet : See Section “BitSet” in *adt*
function KeyInTable : See Section “Table” in *adt*
function NewBitSet : See Section “BitSet” in *adt*
function NewInput : See Section “CmdLineIncl” in *input*
function NewInput : See Section “Include” in *input*
function NextElemInBitSet : See Section “BitSet” in *adt*
function NullIS : See Section “IntSet” in *adt*
function P_outstr : See Section “OutStr” in *output*
function PrintBitSet : See Section “BitSet” in *adt*
function PrintElemsBitSet : See Section “BitSet” in *adt*
function SetMap : See Section “Map” in *adt*
function SingleIS : See Section “IntSet” in *adt*

function SubElemFromBitSet : See Section “BitSet” in *adt*
function SubIS : See Section “IntSet” in *adt*
function SubtractBitSet : See Section “BitSet” in *adt*
function SubtractFromBitSet : See Section “BitSet” in *adt*
function UnionToBitSet : See Section “BitSet” in *adt*
function UniteBitSet : See Section “BitSet” in *adt*
function UniteIS : See Section “IntSet” in *adt*
function outstr : See Section “OutStr” in *output*
function AddIdn : See Section “Bindings” in *name*
function AddToOrderedSet : See Section “List” in *adt*
function AddToSet : See Section “List” in *adt*
function App : See Section “List” in *adt*
function AppEl : See Section “List” in *adt*
function BindIdn : See Section “Bindings” in *name*
function BindInScope : See Section “Bindings” in *name*
function BindKey : See Section “Bindings” in *name*
function BindKeyInScope : See Section “Bindings” in *name*
function BindingInEnv : See Section “Lookup” in *name*
function BindingInScope : See Section “Lookup” in *name*
function Comp : See Section “List” in *adt*
function Cons : See Section “List” in *adt*
function Copy : See Section “List” in *adt*
function DefineIdn : See Section “Bindings” in *name*
function ElemIn : See Section “List” in *adt*
function Filter : See Section “LidoList” in *adt*
function Finl : See Section “List” in *adt*
function Head : See Section “List” in *adt*
function InheritClass : See Section “Inheritance” in *name*
function Inheritsfrom : See Section “Inheritance” in *name*
function InsertAfter : See Section “List” in *adt*
function IthElem : See Section “List” in *adt*
function KeyInEnv : See Section “Lookup” in *name*
function KeyInScope : See Section “Lookup” in *name*
function Length : See Section “List” in *adt*
function Map : See Section “List” in *adt*
function MapFct : See Section “List” in *adt*
function NextInhBinding : See Section “Hidden” in *name*
function NextInhKey : See Section “Hidden” in *name*
function OrderedInsert : See Section “List” in *adt*
function OverridesBinding : See Section “Hidden” in *name*
function PreDefine : See Section “Predefined Identifiers” in *name*
function PreDefineSym : See Section “Predefined Identifiers” in *name*
function RefEndApp : See Section “List” in *adt*
function RefEndCons : See Section “List” in *adt*
function Single : See Section “List” in *adt*
function Sum : See Section “List” in *adt*
function SumFct : See Section “List” in *adt*

function Tail : See Section “List” in *adt*
function application : See Section “Attribution” in *novice*
function calls : See Section “Simple Expressions” in *lidoref*
function signature : See Section “Calls” in *ptg*
function signature : See Section “Patterns” in *ptg*
function signature : See Section “Typed” in *ptg*
function type : See Section “LidoList” in *adt*
function, node construction : See Section “Construction” in *tp*
functions, printing : See Section “Text generation” in *novice*
functions, text generation : See Section “Text generation” in *novice*
fwGen : See Section “fwGen” in *pp*
fwHtml : See Section “fwHtml” in *pp*
fwTex : See Section “fwTex” in *pp*
g : See Section “Expert” in *info*
gdb : See Section “Debugging” in *pp*
gencode : See Section “gencode” in *pp*
general format : See Section “Command line format” in *clp*
generated files : See Section “gencode” in *pp*
generated files : See Section “help” in *pp*
generated program, characteristics : See Section “Overview” in *novice*
generated trees : See Section “Bottom-up Evaluation” in *news4.1*
generation of program text : See Section “Text generation” in *novice*
generation, parser : See Section “parsable” in *pp*
generic module : See Section “Entities” in *novice*
generic module : See Section “Instantiation” in *novice*
generic module : See Section “Output” in *novice*
generic module : See Section “Specifications” in *novice*
generic modules, naming instances : See Section “instance” in *pp*
generic modules, user-constructed : See Section “inst” in *pp*
global interprocess communication : See Section “Implement” in *sysadmin*
grammar : See Section “Specifications” in *lex*
grammar : See Section “Phrases” in *syntax*
grammar changes to resolve ambiguity : See Section “Changes” in *syntax*
grammar development : See Section “Structure” in *novice*
grammar listing : See Section “Grammar” in *clp*
grammar root : See Section “Grammar Root Symbol” in *news4.1*
grammar rule : See Section “Syntactic analysis” in *novice*
growth : See Section “obstack” in *lib*
halt of the parser : See Section “Parsing” in *syntax*
hash : See Section “Hash” in *problems*
head of coercion sequence : See Section “Coercion Sequences” in *oil*
header file : See Section “source” in *pp*
header files : See Section “Header Files” in *mon*
help : See Section “help” in *pp*
help derivation : See Section “Errors” in *comptrees*
help request to Eli : See Section “Session” in *novice*
history : See Section “parex” in *novice*

hypertext : See [Section “help”](#) in *pp*
identifier : See [Section “Property storage”](#) in *novice*
identifier : See [Section “Example”](#) in *oil*
identifier : See [Section “Identifiers”](#) in *ptg*
identifier : See [Section “Specification”](#) in *tp*
identifier declarations, scope issues : See [Section “Scope”](#) in *syntax*
identifier occurrence : See [Section “Usage”](#) in *prop*
identifier output : See [Section “fold”](#) in *pp*
identifier roles : See [Section “Preconditions”](#) in *name*
identifier table : See [Section “Specifications”](#) in *novice*
identifier, form of : See [Section “Nonterminals”](#) in *syntax*
identifier, in odin-expressions : See [Section “Products”](#) in *novice*
identifiers : See [Section “Overall Structure”](#) in *lidoref*
identifiers : See [Section “Syntax”](#) in *lidoref*
identifiers : See [Section “Data”](#) in *ptg*
identifiers : See [Section “Syntax”](#) in *ptg*
identifiers : See [Section “Typed”](#) in *ptg*
idn.c : See [Section “identifier”](#) in *lib*
idn.h : See [Section “identifier”](#) in *lib*
ignore : See [Section “ignore”](#) in *pp*
illegal operator : See [Section “Validation”](#) in *oil*
implementation : See [Section “Implementation”](#) in *tp*
implementation of C entities : See [Section “Implementing”](#) in *comptrees*
implementing monitoring interfaces : See [Section “Implementing Interfaces”](#) in *mon*
implicit definitions : See [Section “Basic Scope Rules”](#) in *name*
impossible coercion : See [Section “Simple Example”](#) in *oil*
include : See [Section “ignore”](#) in *pp*
include : See [Section “Specification”](#) in *tp*
include directive : See [Section “Specifications”](#) in *novice*
include files : See [Section “Include Files”](#) in *fw*
include files : See [Section “Header Files”](#) in *mon*
inconsistencies : See [Section “Diagnostics”](#) in *pp*
incremental design : See [Section “Design Hints”](#) in *oil*
indentation : See [Section “Macro Expansion”](#) in *fw*
indentation : See [Section “BlockPrint”](#) in *output*
indentation : See [Section “Indent”](#) in *output*
indentation : See [Section “PrettyPrint”](#) in *output*
indentation : See [Section “Calls”](#) in *ptg*
indentation, blank : See [Section “Macro Expansion”](#) in *fw*
indentation, none : See [Section “Macro Expansion”](#) in *fw*
index : See [Section “Attributes”](#) in *lidoref*
index : See [Section “Productions”](#) in *lidoref*
index : See [Section “Terminal Access”](#) in *lidoref*
indexed insertion points, order of : See [Section “treesep”](#) in *idem*
inheritance : See [Section “Inheritance”](#) in *comptrees*
inheritance : See [Section “LIGA Files”](#) in *comptrees*
inheritance : See [Section “Inheritance of Computations”](#) in *lidoref*

inheritance : See [Section “AlgInh”](#) in *name*
inheritance : See [Section “CInh”](#) in *name*
inheritance : See [Section “Environment Module”](#) in *name*
inheritance : See [Section “Instantiation”](#) in *novice*
inheritance of scopes : See [Section “Inheritance of Scopes”](#) in *name*
inheritance relation : See [Section “Inheritance of Computations”](#) in *lidoref*
inherited : See [Section “Value”](#) in *comptrees*
inherited : See [Section “Attributes”](#) in *lidoref*
inherited : See [Section “Types and Classes of Attributes”](#) in *lidoref*
inherited attribute : See [Section “Symbol Specifications”](#) in *lidoref*
init event type : See [Section “Breakpoints”](#) in *mon*
initBuf : See [Section “source”](#) in *lib*
initialisation : See [Section “Breakpoints”](#) in *mon*
initialisation : See [Section “User Initialisation”](#) in *mon*
initialization : See [Section “Specifications”](#) in *novice*
initialization grammar : See [Section “Initializations”](#) in *deftbl*
input : See [Section “Input parameters”](#) in *clp*
input file : See [Section “Default behaviour”](#) in *clp*
input file inclusion : See [Section “CmdLineIncl”](#) in *input*
input file inclusion : See [Section “Include”](#) in *input*
input file insertion : See [Section “top”](#) in *input*
input file line length : See [Section “Input Length”](#) in *fw*
input file line length : See [Section “Introduction”](#) in *fw*
input parameter : See [Section “Input parameter access”](#) in *clp*
input parameter : See [Section “Input parameters”](#) in *clp*
input parameters : See [Section “CLP”](#) in *news4.0*
input text display : See [Section “Input Text Display”](#) in *mon*
inputs : See [Section “inputs”](#) in *pp*
inserting EOL markers : See [Section “Inserting EOL”](#) in *fw*
inserting arbitrary characters into text : See [Section “Inserting Arbitrary”](#) in *fw*
inserting control characters into text : See [Section “Inserting Arbitrary”](#) in *fw*
inserting special character into text : See [Section “Inserting Special”](#) in *fw*
insertion point : See [Section “Indexed”](#) in *ptg*
insertion point : See [Section “Patterns”](#) in *ptg*
insertion point : See [Section “Typed”](#) in *ptg*
insertion points : See [Section “Computed Subtrees”](#) in *lidoref*
insertion points : See [Section “Predefined Entities”](#) in *lidoref*
inst : See [Section “inst”](#) in *pp*
installation : See [Section “Install”](#) in *sysadmin*
installing Eli : See [Section “EliInstall”](#) in *sysadmin*
installing packages : See [Section “PkgInstall”](#) in *sysadmin*
instance : See [Section “instance”](#) in *pp*
instance names : See [Section “Instantiation”](#) in *modlib*
instance parameter : See [Section “Instantiation”](#) in *modlib*
instanciating modules from .fw-files : See [Section “Instanciation of Modules with Filenames”](#) in *news4.1*
instantiation : See [Section “Instantiation”](#) in *modlib*

instantiation : See Section “Instantiation” in *novice*
instantiation : See Section “inst” in *pp*
instantiation : See Section “Usage” in *prop*
int : See Section “Value options” in *clp*
int : See Section “Data” in *ptg*
int : See Section “Typed” in *ptg*
integer values : See Section “Value options” in *clp*
interactive : See Section “Debugging” in *pp*
interactive : See Section “help” in *pp*
interactive Eli session : See Section “Session” in *novice*
interface file : See Section “Introduction” in *ptg*
interface file : See Section “Predefined” in *ptg*
interface file : See Section “Predefined” in *tp*
intermediate products : See Section “Derivation” in *pp*
internal code : See Section “printtokens” in *pp*
interprocess communication, debugging : See Section “Implement” in *sysadmin*
intrinsic value : See Section “printtokens” in *pp*
ints : See Section “Value options” in *clp*
invalid key : See Section “Keys” in *deftbl*
invoking Noosa : See Section “Invoking Noosa” in *mon*
invoking type-`gnrc` scripts : See Section “inst” in *pp*
ipc : See Section “Communicate” in *sysadmin*
iteration : See Section “Iterations” in *lidoref*
joined value options : See Section “Joined value options” in *clp*
joinedto : See Section “Joined value options” in *clp*
key : See Section “Keys” in *deftbl*
key : See Section “top” in *name*
key : See Section “Property storage” in *novice*
key : See Section “top” in *prop*
keyword : See Section “kwd” in *pp*
kinds of unparser : See Section “Unparsings” in *idem*
known keys : See Section “Initializations” in *deftbl*
kwd : See Section “kwd” in *pp*
l : See Section “Help-Adv” in *info*
last modification : See Section “test” in *pp*
latex : See Section “FunnelWeb” in *news4.2*
latex : See Section “PDF” in *news4.4*
latex2html : See Section “FunnelWeb” in *news4.2*
layout : See Section “Use” in *idem*
layout : See Section “treesep” in *idem*
leaf : See Section “Structure” in *tp*
leave event : See Section “Time Profiles” in *mon*
left-associative : See Section “Association” in *syntax*
length of a line : See Section “source” in *lib*
letters, case of : See Section “fold” in *pp*
lexerr : See Section “Available Processors” in *lex*
lexerr : See Section “Lexical Analysis” in *news4.2*

lexical analysis : See Section “Lexical Structure” in *mon*
lexical analysis : See Section “Lexical analysis” in *novice*
lexical analyzer : See Section “Terminals” in *syntax*
lexical structure : See Section “Lexical Structure” in *mon*
library : See Section “inst” in *pp*
library functions : See Section “Library” in *oil*
library routine : See Section “source” in *pp*
liga protocol : See Section “LIGA” in *news4.5*
line comments : See Section “Syntax” in *lidoref*
line width : See Section “BlockPrint” in *output*
line width : See Section “PrettyPrint” in *output*
linear lists : See Section “LidoList” in *adt*
linear lists : See Section “List” in *adt*
linked list : See Section “Repeated options” in *clp*
lisedit : See Section “error” in *lib*
list : See Section “Sequences” in *ptg*
list functions : See Section “List” in *adt*
list object, output : See Section “Interactive” in *novice*
list separators : See Section “Separators” in *syntax*
literal : See Section “Atoms” in *novice*
literal : See Section “Specifications” in *novice*
literal : See Section “Syntactic analysis” in *novice*
literal chain rules : See Section “Chain Rules” in *syntax*
literal chain rules and Idem : See Section “Completion” in *syntax*
literal chain rules, preserving : See Section “Completion” in *syntax*
literal chain rules, preserving : See Section “Mapping Chain Rules” in *syntax*
literal terminal : See Section “Tree” in *comptrees*
literal terminals : See Section “Productions” in *lidoref*
literal terminals, reconstruction : See Section “idem” in *idem*
literal terminals, reconstruction : See Section “idemsep” in *idem*
literal, form of : See Section “Terminals” in *syntax*
literals : See Section “Simple Expressions” in *lidoref*
literals, abstraction from : See Section “Abstraction” in *syntax*
literate programming : See *fw*
local interprocess communication : See Section “Implement” in *sysadmin*
long : See Section “Typed” in *ptg*
longest match : See Section “Ambiguity” in *lex*
lower case : See Section “fold” in *pp*
lower computation : See Section “Terminal Access” in *lidoref*
lower computations : See Section “Symbol Specifications” in *lidoref*
lower context : See Section “Tree” in *comptrees*
lower context : See Section “Attributes” in *lidoref*
lower context : See Section “Instantiation” in *novice*
m : See Section “Help-M” in *info*
macro attributes : See Section “Macros” in *fw*
macro definition : See Section “Specifications” in *novice*
macro names : See Section “Names” in *fw*

macro parameter delimiting : See Section “Macro Calls” in *fw*
mail : See Section “Complaining” in *info*
main program : See Section “SO” in *news4.8*
main program, omitting : See Section “nomain” in *pp*
maintainer : See Section “Complaining” in *info*
makefile : See Section “source” in *pp*
malloc : See Section “obstack” in *lib*
mapping : See Section “Tree Construction” in *comptrees*
maximum input line length : See Section “Input Length” in *fw*
maximum input line length : See Section “Introduction” in *fw*
maximum output line length : See Section “Introduction” in *fw*
maximum output line length : See Section “Macro Expansion” in *fw*
maximum product file line length : See Section “Macro Expansion” in *fw*
meaning of a construct : See Section “Meaning” in *syntax*
memory exhaustion : See Section “obstack” in *lib*
menu : See Section “Help-M” in *info*
message : See Section “error” in *lib*
message : See Section “Strings” in *problems*
messages : See Section “Messages” in *mon*
messages : See Section “Message” in *problems*
minus : See Section “Classes” in *lex*
missing definition : See Section “Basic Scope Rules” in *name*
mkidn : See Section “identifier” in *lib*
mkidn : See Section “fold” in *pp*
mkstr : See Section “Attribution” in *novice*
modification specification : See Section “Modifiers” in *syntax*
modification, checking for : See Section “test” in *pp*
modifications to resolve ambiguity : See Section “Modifiers” in *syntax*
module library : See Section “LIGA Files” in *comptrees*
module, block print : See Section “treeseq” in *idem*
module, generic : See Section “Entities” in *novice*
module, separator : See Section “idemsep” in *idem*
mon : See Section “Monitoring” in *pp*
mon : See Section “monitor” in *pp*
mondbx : See Section “Monitoring” in *news4.0*
mongdb : See Section “Monitoring” in *news4.0*
mongdb : See Section “Monitoring” in *pp*
mongdb : See Section “monitor” in *pp*
monitor : See Section “monitor” in *pp*
monitoring : See Section “Monitoring” in *news4.0*
monitoring : See Section “Products” in *novice*
monitoring : See Section “Monitoring” in *pp*
monitoring : See Section “monitor” in *pp*
monitoring database : See Section “Database” in *mon*
monitoring database : See Section “Implementing Interfaces” in *mon*
monitoring interface : See Section “Monitoring Interfaces” in *mon*
monitoring interface implementation : See Section “Implementing Interfaces” in *mon*

monitoring non-standard types : See Section “Non-standard types” in *mon*
monitoring support : See Section “Monitoring Support” in *mon*
mouse buttons : See Section “Invoking Noosa” in *mon*
move by the parser : See Section “Parsing” in *syntax*
move determined by input symbol : See Section “Parsing” in *syntax*
multiple caches : See Section “Cache” in *novice*
multiple inheritance : See Section “Inheritance of Computations” in *lidoref*
multiple operator definition : See Section “OPER” in *oil*
multiple option strings : See Section “Multiple option strings” in *clp*
multiple option strings : See Section “Clp” in *news4.1*
multiple property definitions : See Section “Declarations” in *deftbl*
n : See Section “Help” in *info*
name analysis : See Section “top” in *name*
name analysis : See Section “inst” in *pp*
name analysis test : See Section “Module for Testing Name Analysis” in *news4.1*
name of a derived object : See Section “Products” in *novice*
name spaces : See Section “Environment Module” in *name*
name, cache : See Section “CacheName” in *sysadmin*
name, of an object : See Section “objects” in *ui*
named terminal : See Section “Tree” in *comptrees*
named terminal : See Section “Tree Construction Functions” in *lidoref*
named terminals : See Section “Outdated Constructs” in *lidoref*
named terminals : See Section “Productions” in *lidoref*
named terminals : See Section “Terminal Access” in *lidoref*
names : See Section “Names” in *fw*
names : See Section “Preconditions” in *name*
names : See Section “Name Functions” in *oil*
names : See Section “Names” in *oil*
names : See Section “Using Names” in *oil*
names, quick : See Section “Quick Names” in *fw*
naming instances of generic modules : See Section “instance” in *pp*
nested ranges : See Section “Preconditions” in *name*
nested regions : See Section “Instantiation” in *novice*
newline : See Section “Output” in *ptg*
newline : See Section “Syntax” in *ptg*
newline character : See Section “Atoms” in *novice*
newline defaults : See Section “White Space” in *lex*
newlines : See Section “source” in *lib*
next : See Section “Help” in *info*
node : See Section “Help” in *info*
node : See Section “Attribution” in *novice*
node : See *syntax*
node : See Section “Rules” in *tp*
node construction function : See Section “Construction” in *tp*
nomain : See Section “SO” in *news4.8*
nomain : See Section “nomain” in *pp*
non-associative : See Section “Association” in *syntax*

non-interactive Eli session : See Section “Session” in *novice*
non-literal terminal symbols : See Section “Combine” in *idem*
non-product files : See Section “Output Files” in *fw*
non-product output files : See Section “FunnelWeb” in *news4.1*
non-standard types, browsing support : See Section “Browsing non-standard types” in *mon*
non-standard types, monitoring : See Section “Non-standard types” in *mon*
nonliteral symbol : See Section “Atoms” in *novice*
nonterminal : See Section “Tree” in *comptrees*
nonterminal : See Section “Productions” in *lidoref*
nonterminal : See Section “Patterns” in *tp*
nonterminal symbol : See Section “Syntactic analysis” in *novice*
nonterminal symbols in a grammar : See Section “Nonterminals” in *syntax*
nonterminal types : See Section “Evaluation” in *tp*
noosa : See Section “Monitoring” in *news4.0*
numbers : See Section “Data” in *ptg*
numstr : See Section “storage” in *lib*
object kind : See Section “Kind” in *prop*
object kind : See Section “KindSet” in *prop*
object name : See Section “objects” in *ui*
object, derived : See Section “Products” in *novice*
object, source : See Section “Products” in *novice*
object-oriented : See Section “Environment Module Enhancements” in *news4.1*
obstack : See Section “DynSpace” in *adt*
obstack : See Section “obstack” in *lib*
obstack_1grow : See Section “obstack” in *lib*
obstack_1grow_fast : See Section “obstack” in *lib*
obstack_alignment_mask : See Section “obstack” in *lib*
obstack_alloc : See Section “obstack” in *lib*
obstack_base : See Section “obstack” in *lib*
obstack_begin : See Section “obstack” in *lib*
obstack_blank : See Section “obstack” in *lib*
obstack_blank_fast : See Section “obstack” in *lib*
obstack_chunk_alloc : See Section “obstack” in *lib*
obstack_chunk_free : See Section “obstack” in *lib*
obstack_chunk_realloc : See Section “obstack” in *lib*
obstack_chunk_size : See Section “obstack” in *lib*
obstack_copy : See Section “obstack” in *lib*
obstack_copy0 : See Section “obstack” in *lib*
obstack_finish : See Section “obstack” in *lib*
obstack_free : See Section “obstack” in *lib*
obstack_grow : See Section “obstack” in *lib*
obstack_grow0 : See Section “obstack” in *lib*
obstack_init : See Section “obstack” in *lib*
obstack_int_grow : See Section “obstack” in *lib*
obstack_int_grow_fast : See Section “obstack” in *lib*
obstack_next_free : See Section “obstack” in *lib*

obstack_object_size : See Section “obstack” in *lib*
obstack_ptr_grow : See Section “obstack” in *lib*
obstack_ptr_grow_fast : See Section “obstack” in *lib*
obstack_room : See Section “obstack” in *lib*
obstack_strcpy : See Section “obstack” in *lib*
odin-expression : See Section “Products” in *novice*
odin-expressions : See Section “objects” in *ui*
omitted chain rule : See Section “Compute” in *idem*
omitted child : See Section “Combine” in *idem*
omitting the main program : See Section “nomain” in *pp*
omitting the parser : See Section “parser” in *pp*
one or more : See Section “Plus” in *lex*
online help : See Section “top” in *mon*
open error format : See Section “Usage options” in *clp*
open error usage : See Section “Usage options” in *clp*
operation : See Section “Operations” in *mon*
operation macros : See Section “Operations” in *deftbl*
operation names : See Section “Declarations” in *deftbl*
operator : See Section “Expression” in *syntax*
operator association : See Section “Association” in *syntax*
operator character : See Section “Quoting” in *lex*
operator definition : See Section “OPER” in *oil*
operator definition : See Section “Statements” in *oil*
operator denotation : See Section “Complex Example” in *oil*
operator denotation : See Section “Example” in *oil*
operator identification : See Section “INDICATION” in *oil*
operator identification : See Section “Simple Example” in *oil*
operator identification : See Section “Statements” in *oil*
operator indication : See Section “Complex Example” in *oil*
operator indication : See Section “Example” in *oil*
operator indication : See Section “Possible Types” in *oil*
operator precedence : See Section “Precedence” in *syntax*
operator precedence, overriding : See Section “idem” in *idem*
operator signature : See Section “Simple Example” in *oil*
operator, in odin-expressions : See Section “Products” in *novice*
operators : See Section “Language” in *oil*
optional : See Section “Query” in *lex*
optional output patterns : See Section “Optional” in *ptg*
options : See Section “Controlling Your Program” in *mon*
order of indexed insertion points : See Section “treesep” in *idem*
order of option specification lines : See Section “Ordering options” in *clp*
order of specification lines : See Section “Positional parameters” in *clp*
ordering of specifications : See Section “Ambiguity” in *lex*
outdated constructs : See Section “LIGA” in *news4.5*
output : See Section “Text generation” in *novice*
output : See Section “output” in *pp*
output file line length : See Section “Introduction” in *fw*

output file line length : See Section “Macro Expansion” in *fw*
output files : See Section “Output Files” in *fw*
output function : See Section “Frame” in *ptg*
output functions : See Section “BlockPrint” in *output*
output functions : See Section “PrettyPrint” in *output*
output of identifiers : See Section “fold” in *pp*
output text structure : See Section “Specifications” in *novice*
output text structure : See Section “Text generation” in *novice*
output to a file : See Section “Interactive” in *novice*
outputs : See Section “outputs” in *pp*
overload resolution : See Section “Specifications” in *novice*
overload resolution : See Section “Subproblems” in *novice*
overload resolution : See Section “Expression” in *syntax*
overloaded operators : See Section “Example” in *oil*
overriding : See Section “Inheritance of Computations” in *lidoref*
overriding PTG patterns : See Section “Changing” in *idem*
overriding computations : See Section “Basic” in *comptrees*
overriding operator precedence : See Section “idem” in *idem*
p : See Section “Help-P” in *info*
package : See Section “Package” in *news4.6*
package : See *sysadmin*
package collections : See Section “Collections” in *sysadmin*
package definition : See Section “treeIn” in *idem*
package library : See Section “installpkg” in *news4.6*
packages, cache : See Section “Capability” in *sysadmin*
packages, installing : See Section “PkgInstall” in *sysadmin*
packages, versioned : See Section “Versioning” in *sysadmin*
pairwise related keys : See Section “Reflex” in *prop*
parameter : See Section “top” in *pp*
parameter type : See Section “LidoList” in *adt*
parameter type : See Section “parameterization expressions” in *ui*
parameter value : See Section “parameterization expressions” in *ui*
parameters : See Section “Breakpoints” in *mon*
parameters of the eli command ‘R’ : See Section “Upgrading” in *faq*
parameters of the eli command ‘r’ : See Section “CacheSize” in *faq*
parameters, -p : See Section “EliCmd” in *news4.6*
parameters, command line : See Section “arg” in *pp*
parsable : See Section “parsable” in *pp*
parser : See Section “Structure” in *novice*
parser : See Section “parser” in *pp*
parser actions : See Section “Actions” in *syntax*
parser construction : See Section “Modifiers” in *syntax*
parser error reports : See Section “Phrases” in *syntax*
parser generation : See Section “parsable” in *pp*
parser generator selection : See Section “parser” in *pp*
parser generators : See Section “Phrase Structure” in *mon*
parser move determined by input symbol : See Section “Parsing” in *syntax*

parser operation : See Section “Parsing” in *syntax*
 parser, omitting : See Section “parser” in *pp*
 parsing : See Section “Phrase Structure” in *mon*
 parsing routine : See Section “Phrases” in *syntax*
 passed through arguments : See Section “Calls” in *ptg*
 pattern : See Section “Patterns” in *ptg*
 pattern function : See Section “Patterns” in *ptg*
 pdl : See Section “Definition” in *deftbl*
 pdl_gen.h : See Section “Interface” in *deftbl*
 period : See Section “Classes” in *lex*
 pgram : See Section “pgram” in *pp*
 phrase : See Section “Syntactic analysis” in *novice*
 phrase : See *syntax*
 phrase structure : See Section “Phrase Structure” in *mon*
 phrase structure : See Section “Specifications” in *novice*
 phrase structure : See Section “Structure” in *novice*
 phrase structure : See Section “Syntactic analysis” in *novice*
 phrase structure : See Section “Phrases” in *syntax*
 phrase structure : See Section “Specification” in *tp*
 plain computation : See Section “Symbol Specifications” in *lidoref*
 plain computations : See Section “Attribute Computations and Plain Computations” in *lidoref*
 plain computations : See Section “Computations” in *lidoref*
 plain computations : See Section “Inheritance of Computations” in *lidoref*
 plain productions, computations : See Section “idemsep” in *idem*
 plain productions, computations : See Section “treeseq” in *idem*
 pointer : See Section “Calls” in *ptg*
 positional : See Section “Positional parameters” in *clp*
 positional parameters : See Section “Command line interface” in *clp*
 positional parameters : See Section “Positional parameters” in *clp*
 positional parameters : See Section “Clp” in *news4.1*
 positionals : See Section “Positional parameters” in *clp*
 possible result types : See Section “Possible Types” in *oil*
 postcondition : See Section “State” in *comptrees*
 postcondition : See Section “Attribute Computations and Plain Computations” in *lidoref*
 postcondition : See Section “CHAIN” in *lidoref*
 postcondition : See Section “Iterations” in *lidoref*
 pre-processor directives : See Section “Definition” in *deftbl*
 precedence levels : See Section “Use” in *idem*
 precedence rules : See Section “Precedence” in *syntax*
 precondition : See Section “State” in *comptrees*
 precondition : See Section “Attribute Computations and Plain Computations” in *lidoref*
 precondition : See Section “CHAIN” in *lidoref*
 precondition : See Section “Dependent Expressions” in *lidoref*
 predefined entities : See Section “Implementing” in *comptrees*
 predefined identifiers : See Section “Predefined Identifiers” in *name*
 predefined macro : See Section “Value” in *comptrees*

predefined symbols : See Section “Canned Descriptions” in *lex*
prefixes : See Section “Ordering options” in *clp*
preserving literal chain rules : See Section “Completion” in *syntax*
preserving literal chain rules : See Section “Mapping Chain Rules” in *syntax*
pretty printing : See Section “PrettyPrint” in *output*
pretty-printer : See Section “Use” in *idem*
previous : See Section “Help” in *info*
primary input file : See Section “Refactor” in *news4.8*
printf : See Section “Introduction” in *ptg*
printing : See Section “Options” in *info*
printing functions : See Section “Text generation” in *novice*
printtokens : See Section “printtokens” in *pp*
problem reporting : See Section “Problems” in *sysadmin*
procedure declaration, scope issues : See Section “Scope” in *syntax*
processor, token : See Section “Attribution” in *novice*
product : See Section “top” in *pp*
product file line length : See Section “Macro Expansion” in *fw*
production : See Section “Tree” in *comptrees*
production : See Section “Rule Specifications” in *lidoref*
production : See Section “Phrases” in *syntax*
productions : See Section “Computed Subtrees” in *lidoref*
productions, LISTOF : See Section “idem” in *idem*
products, relationships among : See Section “Derivation” in *pp*
program arguments : See Section “Controlling Your Program” in *mon*
program options : See Section “Controlling Your Program” in *mon*
program text generation : See Section “Specifications” in *novice*
program text generation : See Section “Text generation” in *novice*
progress in recognizing a phrase : See Section “Parsing” in *syntax*
property : See Section “top” in *prop*
property From : See Section “Reflex” in *prop*
property IsType : See Section “Types” in *type*
property KindSet : See Section “KindSet” in *prop*
property Kind : See Section “Kind” in *prop*
property ObjNo : See Section “ObjCnt” in *prop*
property Scope : See Section “ScopeProp” in *name*
property To : See Section “Reflex” in *prop*
property Defer : See Section “Defer” in *prop*
property Line : See Section “Name Analysis Test” in *name*
property Scope : See Section “Inheritance of Scopes” in *name*
property declaration : See Section “Properties” in *deftbl*
property definition : See Section “Specifications” in *novice*
property definition language : See Section “Definition” in *deftbl*
property name : See Section “Properties” in *deftbl*
property specification : See Section “Declarations” in *deftbl*
property type : See Section “Properties” in *deftbl*
prtidnv : See Section “identifier” in *lib*
prtstcon : See Section “storage” in *lib*

ps : See [Section “ps”](#) in *pp*
ptg-gen.c : See [Section “Introduction”](#) in *ptg*
ptg-gen.h : See [Section “Introduction”](#) in *ptg*
ptg-gen.h : See [Section “Predefined”](#) in *ptg*
q : See [Section “Help-Q”](#) in *info*
query : See [Section “Property storage”](#) in *novice*
query and update operations : See [Section “Interface”](#) in *deftbl*
query operation : See [Section “Query”](#) in *deftbl*
quick names : See [Section “Quick Names”](#) in *fw*
quitting Noosa : See [Section “Invoking Noosa”](#) in *mon*
quote : See [Section “Quoting”](#) in *lex*
r : See [Section “Examples”](#) in *info*
r parameter of the eli command : See [Section “CacheSize”](#) in *faq*
range : See [Section “Classes”](#) in *lex*
range : See [Section “Usage”](#) in *prop*
ranked alphabet : See [Section “Patterns”](#) in *tp*
ranked alphabet : See [Section “Rules”](#) in *tp*
ranked alphabet : See [Section “Structure”](#) in *tp*
rc file : See [Section “User Initialisation”](#) in *mon*
realloc : See [Section “obstack”](#) in *lib*
reconstruction of literal terminals : See [Section “idem”](#) in *idem*
reconstruction of literal terminals : See [Section “idemsep”](#) in *idem*
redo : See [Section “redo”](#) in *pp*
reduce move by the parser : See [Section “Parsing”](#) in *syntax*
reduce-reduce conflict : See [Section “Parsing”](#) in *syntax*
reduce-reduce solved by @ modification : See [Section “At”](#) in *syntax*
referred set : See [Section “CONSTITUENT\(S\)”](#) in *lidoref*
referred set : See [Section “INCLUDING”](#) in *lidoref*
referto : See [Section “referto”](#) in *pp*
referto parameter : See [Section “Instantiation”](#) in *modlib*
refillBuf : See [Section “source”](#) in *lib*
region : See [Section “Entities”](#) in *novice*
regions, nested : See [Section “Instantiation”](#) in *novice*
regular expression : See [Section “Regular Expressions”](#) in *lex*
relationship : See [Section “Property storage”](#) in *novice*
remote dependencies : See [Section “Attribute Class Conflict”](#) in *comptrees*
remote dependencies : See [Section “Including”](#) in *comptrees*
remote dependencies : See [Section “Remote”](#) in *comptrees*
repairing syntactic errors : See [Section “Phrases”](#) in *syntax*
repeated boolean options : See [Section “Repeated options”](#) in *clp*
repeated boolean options : See [Section “Clp”](#) in *news4.1*
repeated options : See [Section “Boolean options”](#) in *clp*
repeated options : See [Section “Joined value options”](#) in *clp*
repeated options : See [Section “Repeated options”](#) in *clp*
repeated options : See [Section “Value options”](#) in *clp*
repetition : See [Section “Plus”](#) in *lex*
reported error in a modification : See [Section “Modifiers”](#) in *syntax*

reports of syntactic errors : See Section “Phrases” in *syntax*
requesting source code, example : See Section “symbex” in *novice*
reset times : See Section “Time Profiles” in *mon*
resolving ambiguity by grammar changes : See Section “Changes” in *syntax*
resolving ambiguity by modifications : See Section “Modifiers” in *syntax*
resolving ambiguity, general methods : See Section “Conflicts” in *syntax*
resolving overloading : See Section “Expression” in *syntax*
restart point : See Section “Error Recovery” in *syntax*
result type : See Section “Coercion Sequences” in *oil*
result type : See Section “Possible Types” in *oil*
reuse of computations : See Section “Abstraction” in *syntax*
right-associative : See Section “Association” in *syntax*
right-hand side : See Section “Right-Hand Side Access” in *news4.1*
role, computational : See Section “Type” in *news4.5*
root : See Section “Attribution” in *novice*
root : See *syntax*
root environment : See Section “AlgScope” in *name*
root environment : See Section “CScope” in *name*
root environment : See Section “Predefined Identifiers” in *name*
root roles : See Section “ModLib” in *news4.1*
root symbol : See Section “Productions” in *lidoref*
root symbol : See Section “Rule Specifications” in *lidoref*
root symbol : See Section “Preconditions” in *name*
rubout : See Section “Help-C-l” in *info*
rule : See Section “Rules” in *tp*
rule attribute type : See Section “Types and Classes of Attributes” in *lidoref*
rule attributes : See Section “Attributes” in *lidoref*
rule computation : See Section “Compute” in *idem*
rule context : See Section “Tree” in *comptrees*
rule name : See Section “Right-Hand Side Access” in *news4.1*
rule specification : See Section “Rule Specifications” in *lidoref*
rule, grammar : See Section “Syntactic analysis” in *novice*
rule, scope : See Section “Property storage” in *novice*
rules, association : See Section “Association” in *syntax*
rules, precedence : See Section “Precedence” in *syntax*
run : See Section “Products” in *novice*
run : See Section “run” in *pp*
run with typical input data : See Section “Testing” in *pp*
running example : See Section “Example” in *modlib*
running example : See Section “top” in *modlib*
running example : See Section “Basic Scope Rules” in *name*
running example : See Section “top” in *name*
running program : See Section “Controlling Your Program” in *mon*
s : See Section “Expert” in *info*
saveidn : See Section “identifier” in *lib*
savestr : See Section “storage” in *lib*
saving handlers : See Section “Breakpoints” in *mon*

saving typing : See Section “Completion” in *info*
scanner : See Section “Tree Construction” in *comptrees*
scanner : See Section “Auxiliary Scanners” in *lex*
scanner : See Section “kwd” in *pp*
scope : See Section “Environment Module” in *name*
scope : See Section “Scope” in *syntax*
scope properties : See Section “Scope Properties” in *name*
scope rules : See Section “Basic Scope Rules” in *name*
scope rules : See Section “top” in *name*
scope rules : See Section “Property storage” in *novice*
screen : See Section “Help-Small-Screen” in *info*
scripts, invocation via **inst** : See Section “inst” in *pp*
section names : See Section “Names” in *fw*
section names : See Section “Structure” in *fw*
selected type : See Section “Possible Types” in *oil*
selecting entities : See Section “Entities” in *deftbl*
semantic brackets : See Section “Brackets” in *syntax*
semantic delimiters : See Section “Brackets” in *syntax*
semantic disambiguation : See Section “Rule Matching” in *syntax*
sentence : See Section “Syntactic analysis” in *novice*
sentence : See *syntax*
sentence symbol : See Section “Phrases” in *syntax*
separate caches : See Section “Cache” in *novice*
separate name space : See Section “AlgScope” in *name*
separate name space : See Section “Basic Scope Rules” in *name*
separate name space : See Section “CScope” in *name*
separator : See Section “CONSTITUENTS” in *ptg*
separator : See Section “Sequences” in *ptg*
separator module, constructing : See Section “idemsep” in *idem*
separator, argument : See Section “Combine” in *idem*
separator, argument : See Section “Compute” in *idem*
separator, argument : See Section “Use” in *idem*
sequence : See Section “Sequences” in *ptg*
sequence : See Section “Phrases” in *syntax*
sequence of coercions : See Section “Coercion Sequences” in *oil*
sequence of coercions : See Section “Simple Example” in *oil*
sequences : See Section “CHAIN” in *ptg*
sequences : See Section “CONSTITUENTS” in *ptg*
server, Odin : See Section “Execution” in *sysadmin*
set of types : See Section “Complex Example” in *oil*
set of types : See Section “Library Types” in *oil*
set of types : See Section “Possible Types” in *oil*
setting special character : See Section “Setting Special” in *fw*
severity of errors : See Section “Instantiation” in *novice*
severity of errors : See Section “Output” in *novice*
shared library : See Section “Foreign” in *news4.8*
shared library : See Section “SO” in *news4.8*

shielding : See Section “CONSTITUENT(S)” in *lidoref*
shift move by the parser : See Section “Parsing” in *syntax*
shift-reduce conflict : See Section “parex” in *novice*
shift-reduce conflict : See Section “Parsing” in *syntax*
shift-reduce solved by \$ modification : See Section “Dollar” in *syntax*
short : See Section “Typed” in *ptg*
showFe : See Section “Void Attributes” in *comptrees*
showFe : See Section “show” in *pp*
showMe : See Section “show” in *pp*
side-effects : See Section “Computations” in *lidoref*
signature : See Section “INDICATION” in *oil*
signature : See Section “OPER” in *oil*
signature : See Section “Signature” in *oil*
signature : See Section “Simple Example” in *oil*
signature, action : See Section “Evaluation” in *tp*
signature, action : See Section “Implementation” in *tp*
significant character sequence : See Section “Atoms” in *novice*
significant character sequence : See Section “Lexical analysis” in *novice*
simple sequence : See Section “Special” in *fw*
single : See Section “Complex Example” in *oil*
single assignment rule : See Section “Value” in *comptrees*
single cache : See Section “Cache” in *novice*
so : See Section “SO” in *news4.8*
so : See Section “so” in *pp*
source : See Section “source” in *pp*
source coordinates : See Section “SetFirst” in *prop*
source language file : See Section “source” in *pp*
source object : See Section “Products” in *novice*
source object : See Section “objects” in *ui*
source text coordinates : See Section “printtokens” in *pp*
source text display : See Section “Input Text Display” in *mon*
source type : See Section “Coercion Sequences” in *oil*
source.h : See Section “source” in *lib*
space : See Section “Help-C-I” in *info*
space defaults : See Section “White Space” in *lex*
special character : See Section “Special” in *fw*
special character, inserting : See Section “Inserting Special” in *fw*
special character, setting : See Section “Setting Special” in *fw*
special sequence : See Section “Special” in *fw*
specification grammar : See Section “Declarations” in *deftbl*
specification grammar : See Section “Operations” in *deftbl*
specification ordering : See Section “Ambiguity” in *lex*
specification types : See Section “Specifications” in *novice*
specifications : See Section “Definition” in *deftbl*
stack of parser states : See Section “Parsing” in *syntax*
standard input : See Section “Command line format” in *clp*
standard input : See Section “Default behaviour” in *clp*

standard input : See Section “Input parameters” in *clp*
standard input : See Section “Input Text Display” in *mon*
standard output : See Section “Interactive” in *novice*
startup file : See Section “Browsing non-standard types” in *mon*
startup file : See Section “User Initialisation” in *mon*
state attribute : See Section “State” in *comptrees*
state dependencies : See Section “State” in *comptrees*
state of the parser : See Section “Parsing” in *syntax*
statements : See Section “CHAIN” in *ptg*
status : See Section “Diagnostics” in *pp*
stdout : See Section “stdout” in *pp*
stop after detecting syntax errors : See Section “Error Recovery” in *syntax*
stopping execution : See Section “Breakpoints” in *mon*
stopping option recognition : See Section “Termination” in *clp*
stostr : See Section “storage” in *lib*
stradd : See Section “strmath” in *lib*
strdivf : See Section “strmath” in *lib*
strdivi : See Section “strmath” in *lib*
string : See Section “Value options” in *clp*
string : See Section “Data” in *ptg*
string : See Section “Syntax” in *ptg*
string : See Section “Typed” in *ptg*
string table : See Section “String value options” in *clp*
string table : See Section “String Table” in *mon*
string table index : See Section “Combine” in *idem*
string values : See Section “Value options” in *clp*
strings : See Section “Value options” in *clp*
strings as numbers : See Section “StrArith” in *problems*
strmath : See Section “strmath” in *lib*
strmult : See Section “strmath” in *lib*
strneg : See Section “strmath” in *lib*
strnorm : See Section “strmath” in *lib*
strnumb : See Section “strmath” in *lib*
strpow : See Section “strmath” in *lib*
strrem : See Section “strmath” in *lib*
strsqrt : See Section “strmath” in *lib*
strsub : See Section “strmath” in *lib*
structural unparser : See Section “tree” in *idem*
structural unparser, deriving : See Section “DeriveTree” in *idem*
structural unparser, structure definition : See Section “DeriveDefn” in *idem*
structured output text : See Section “Specifications” in *novice*
structured output text : See Section “Text generation” in *novice*
sub-tree, computed : See Section “Multiple” in *idem*
subproblem : See Section “Subproblems” in *novice*
subproblem descriptions : See Section “Specifications” in *novice*
support library : See Section “Library” in *oil*
suppressing EOL markers : See Section “Suppressing End of Line” in *fw*

symbol : See [Section “Tree”](#) in *comptrees*
symbol : See [Section “Syntactic analysis”](#) in *novice*
symbol attribution : See [Section “Instantiation”](#) in *novice*
symbol computation : See [Section “Symbols”](#) in *comptrees*
symbol specification : See [Section “Symbol Specifications”](#) in *lidoref*
symbol, nonterminal : See [Section “Syntactic analysis”](#) in *novice*
symbol, terminal : See [Section “Syntactic analysis”](#) in *novice*
syntactic analysis : See [Section “Syntactic analysis”](#) in *novice*
syntactic error reports during parsing : See [Section “Phrases”](#) in *syntax*
syntax errors, stop after detecting : See [Section “Error Recovery”](#) in *syntax*
synthesized : See [Section “Value”](#) in *comptrees*
synthesized : See [Section “Attributes”](#) in *lidoref*
synthesized : See [Section “Types and Classes of Attributes”](#) in *lidoref*
synthesized attribute : See [Section “Symbol Specifications”](#) in *lidoref*
system documentation : See [Section “Documentation”](#) in *novice*
tOilArgSig : See [Section “Signature Constructor”](#) in *oil*
tab character : See [Section “Atoms”](#) in *novice*
tab defaults : See [Section “White Space”](#) in *lex*
tail of coercion sequence : See [Section “Coercion Sequences”](#) in *oil*
target : See [Section “Session”](#) in *novice*
target : See [Section “Odinfile”](#) in *ui*
template : See [Section “Output”](#) in *novice*
template : See [Section “Text generation”](#) in *novice*
termcode.h : See [Section “Building Processors”](#) in *lex*
terminal : See [Section “Tree”](#) in *comptrees*
terminal : See [Section “Tree Construction”](#) in *comptrees*
terminal : See [Section “Value”](#) in *comptrees*
terminal : See [Section “Productions”](#) in *lidoref*
terminal : See [Section “Tree Construction Functions”](#) in *lidoref*
terminal : See [Section “Terminals”](#) in *news4.0*
terminal : See [Section “terminals in LISTOF productions”](#) in *news4.0*
terminal : See [Section “Rules”](#) in *tp*
terminal EOF to terminate text : See [Section “At”](#) in *syntax*
terminal access : See [Section “Right-Hand Side Access”](#) in *news4.1*
terminal attribute : See [Section “LIGA”](#) in *news4.5*
terminal symbol : See [Section “Atoms”](#) in *novice*
terminal symbol : See [Section “Syntactic analysis”](#) in *novice*
terminal symbol, non-literal : See [Section “Combine”](#) in *idem*
terminal symbol, overriding : See [Section “Combine”](#) in *idem*
terminal symbols in a grammar : See [Section “Terminals”](#) in *syntax*
terminal, use in a modification : See [Section “Modifiers”](#) in *syntax*
terminals : See [Section “Outdated Constructs”](#) in *lidoref*
terminals : See [Section “Preconditions”](#) in *name*
termination string : See [Section “Termination”](#) in *clp*
terminator : See [Section “Termination”](#) in *clp*
test : See [Section “test”](#) in *pp*
test output : See [Section “Name Analysis Test”](#) in *name*

tex : See [Section “PDF”](#) in *news4.4*
texinfo : See [Section “Introduction”](#) in *fw*
texinfo : See [Section “PDF”](#) in *news4.4*
text character : See [Section “Quoting”](#) in *lex*
text fragment : See [Section “Text generation”](#) in *novice*
text generation function : See [Section “Output”](#) in *novice*
text generation function : See [Section “Text generation”](#) in *novice*
textual unparser : See [Section “idem”](#) in *idem*
textual unparser, deriving : See [Section “DeriveIdem”](#) in *idem*
time profile : See [Section “Time Profiles”](#) in *mon*
timing : See [Section “Time Profiles”](#) in *mon*
token processor : See [Section “Attribution”](#) in *novice*
token processor compatibility : See [Section “Combine”](#) in *idem*
token processors : See [Section “Interface”](#) in *news4.7*
tokens : See [Section “Syntax”](#) in *ptg*
tool command language : See [Section “Breakpoints”](#) in *mon*
tp_gen.h : See [Section “Predefined”](#) in *tp*
tracing events : See [Section “Tracing Events”](#) in *mon*
tree computations : See [Section “Mapping”](#) in *syntax*
tree fragment : See [Section “Fragment”](#) in *tp*
tree grammar : See [Section “Tree”](#) in *comptrees*
tree grammar : See [Section “Tree Construction”](#) in *comptrees*
tree grammar : See [Section “CHAIN”](#) in *lidoref*
tree grammar : See [Section “CONSTITUENT\(S\)”](#) in *lidoref*
tree grammar : See [Section “Computed Subtrees”](#) in *lidoref*
tree grammar : See [Section “INCLUDING”](#) in *lidoref*
tree grammar : See [Section “Productions”](#) in *lidoref*
tree grammar : See [Section “Rule Specifications”](#) in *lidoref*
tree grammar : See [Section “Symbol Specifications”](#) in *lidoref*
tree grammar : See [Section “Preconditions”](#) in *name*
tree grammar, partial : See [Section “Partial”](#) in *idem*
tree node : See [Section “Node”](#) in *tp*
tree structure : See [Section “Tree”](#) in *comptrees*
tree structure : See [Section “Structure”](#) in *fw*
tree structure : See [Section “Specifications”](#) in *novice*
tree structure, meaning of : See [Section “Structure”](#) in *novice*
tree symbol computation : See [Section “Compute”](#) in *idem*
trivial chain rules : See [Section “Chain Rules”](#) in *syntax*
type : See [Section “Attribute Computations and Plain Computations”](#) in *lidoref*
type : See [Section “CONSTITUENT\(S\)”](#) in *lidoref*
type : See [Section “INCLUDING”](#) in *lidoref*
type : See [Section “Types and Classes of Attributes”](#) in *lidoref*
type BitSet : See [Section “BitSet”](#) in *adt*
type IntSet : See [Section “IntSet”](#) in *adt*
type NODEPTR : See [Section “Computed Subtrees”](#) in *lidoref*
type VOID : See [Section “Attribute Computations and Plain Computations”](#) in *lidoref*
type Binding : See [Section “AlgScope”](#) in *name*

type Binding : See [Section “Basic Scope Rules”](#) in *name*
 type Binding : See [Section “CScope”](#) in *name*
 type Binding : See [Section “Types”](#) in *name*
 type Environment : See [Section “Types”](#) in *name*
 type InheritPtr : See [Section “Types”](#) in *name*
 type analysis : See [Section “Type”](#) in *news4.5*
 type balancing : See [Section “Possible Types”](#) in *oil*
 type denotation : See [Section “Example”](#) in *oil*
 type denotation : See [Section “Possible Types”](#) in *oil*
 type set : See [Section “Complex Example”](#) in *oil*
 type set : See [Section “Language”](#) in *oil*
 type set : See [Section “Library Types”](#) in *oil*
 type set : See [Section “Statements”](#) in *oil*
 type set definition : See [Section “SET”](#) in *oil*
 type signature : See [Section “Signature”](#) in *oil*
 type, of a parameter in an odin-expression : See [Section “parameterization expressions”](#) in *ui*
 type-fw : See [Section “fwTex”](#) in *pp*
 type-gnrc files : See [Section “inst”](#) in *pp*
 type-‘FINL.phi’ file : See [Section “Specifications”](#) in *novice*
 type-‘HEAD.phi’ file : See [Section “Output”](#) in *novice*
 type-‘HEAD.phi’ file : See [Section “Specifications”](#) in *novice*
 type-‘INIT.phi’ file : See [Section “Specifications”](#) in *novice*
 type-‘clp’ file : See [Section “Specifications”](#) in *novice*
 type-‘con’ file : See [Section “Specifications”](#) in *novice*
 type-‘con’ file format : See [Section “Grammars”](#) in *syntax*
 type-‘con’ file, example : See [Section “Con”](#) in *syntax*
 type-‘con’ file, purpose : See [Section “Notation”](#) in *syntax*
 type-‘ctl’ file : See [Section “Specifications”](#) in *novice*
 type-‘dapto’ file format : See [Section “Dapto Grammar”](#) in *mon*
 type-‘delit’ file : See [Section “Specifications”](#) in *novice*
 type-‘eta’ file : See [Section “Specifications”](#) in *novice*
 type-‘fw’ file : See [Section “Generation”](#) in *novice*
 type-‘fw’ file : See [Section “Specifications”](#) in *novice*
 type-‘gla’ file : See [Section “Specifications”](#) in *novice*
 type-‘gnrc’ file : See [Section “Specifications”](#) in *novice*
 type-‘lido’ file : See [Section “Specifications”](#) in *novice*
 type-‘map’ file, example : See [Section “Sym”](#) in *syntax*
 type-‘map’ file, format : See [Section “Grammars”](#) in *syntax*
 type-‘oil’ file : See [Section “Specifications”](#) in *novice*
 type-‘pdl’ file : See [Section “Specifications”](#) in *novice*
 type-‘perr’ file, format : See [Section “Grammars”](#) in *syntax*
 type-‘phi’ file : See [Section “Specifications”](#) in *novice*
 type-‘ptg’ file : See [Section “Specifications”](#) in *novice*
 type-‘specs’ file : See [Section “Specifications”](#) in *novice*
 type-‘str’ file : See [Section “Specifications”](#) in *novice*
 type-‘sym’ file : See [Section “Specifications”](#) in *novice*

types : See [Section “Language”](#) in *oil*
types, attribute : See [Section “Evaluation”](#) in *tp*
types, nonterminal : See [Section “Evaluation”](#) in *tp*
types, of input specification : See [Section “Specifications”](#) in *novice*
typesetter : See [Section “Introduction”](#) in *fw*
typesetter : See [Section “Target”](#) in *fw*
typesetter directives : See [Section “Target”](#) in *fw*
typical use of Eli : See [Section “Example”](#) in *novice*
typing less : See [Section “Completion”](#) in *info*
u : See [Section “Help-FOO”](#) in *info*
ub1 : See [Section “Hash”](#) in *problems*
ub4 : See [Section “Hash”](#) in *problems*
unique pattern names : See [Section “Patterns”](#) in *ptg*
uniqueness of property names : See [Section “Properties”](#) in *deftbl*
unknown value : See [Section “Non-standard types”](#) in *mon*
unparser derivation : See [Section “Derivation”](#) in *idem*
unparser generation : See [Section “Unparsing”](#) in *news4.1*
unparser generator, available specifications : See [Section “treeIng”](#) in *idem*
unparser generator, specifying : See [Section “DeriveTree”](#) in *idem*
unparser, kinds : See [Section “Unparsings”](#) in *idem*
unparser, structural : See [Section “tree”](#) in *idem*
unparser, textual : See [Section “idem”](#) in *idem*
unparsers, combining : See [Section “Multiple”](#) in *idem*
up : See [Section “Help”](#) in *info*
update : See [Section “Property storage”](#) in *novice*
update operation : See [Section “Update”](#) in *deftbl*
upgrading : See [Section “Upgrading”](#) in *faq*
upper case : See [Section “fold”](#) in *pp*
upper computations : See [Section “Symbol Specifications”](#) in *lidoref*
upper context : See [Section “Tree”](#) in *comptrees*
upper context : See [Section “Attributes”](#) in *lidoref*
usage message : See [Section “Documentation”](#) in *clp*
usage message : See [Section “Positional parameter access”](#) in *clp*
usage message : See [Section “Usage options”](#) in *clp*
user initialisation : See [Section “User Initialisation”](#) in *mon*
user modules : See [Section “Instantiation”](#) in *modlib*
user supplied function : See [Section “Calls”](#) in *ptg*
user-defined actions during parsing : See [Section “Actions”](#) in *syntax*
v : See [Section “Examples”](#) in *info*
valid operator : See [Section “Validation”](#) in *oil*
validating a coercion sequence : See [Section “Coercion Sequences”](#) in *oil*
validating operator identification : See [Section “Validation”](#) in *oil*
value : See [Section “Value”](#) in *comptrees*
value context : See [Section “Attribute Computations and Plain Computations”](#) in *lidoref*
value context : See [Section “CONSTITUENT\(S\)”](#) in *lidoref*
value context : See [Section “Simple Expressions”](#) in *lidoref*
value dependencies : See [Section “Value”](#) in *comptrees*

value options : See [Section “Value options”](#) in *clp*
value options with or without spacing : See [Section “Clp”](#) in *news4.1*
value, of a parameter in an odin-expression : See [Section “parameterization expressions”](#) in *ui*
variable entities : See [Section “Entities”](#) in *deftbl*
variables : See [Section “Odinrc”](#) in *faq*
versioned packages : See [Section “Versioning”](#) in *sysadmin*
virtual target : See [Section “Session”](#) in *novice*
virtual target : See [Section “Odinfile”](#) in *ui*
visit-sequence : See [Section “Visit-sequences”](#) in *gorto*
visitprocs messages : See [Section “C-Errors”](#) in *comptrees*
vskip : See [Section “Formatting”](#) in *fw*
warn : See [Section “Products”](#) in *novice*
warning : See [Section “Diagnostics”](#) in *pp*
warning : See [Section “warning”](#) in *pp*
warning message : See [Section “warning”](#) in *pp*
warning messages : See [Section “Errors”](#) in *comptrees*
weave : See [Section “fwTex”](#) in *pp*
white space : See [Section “Atoms”](#) in *novice*
white space : See [Section “Identifiers”](#) in *ptg*
white space : See [Section “Patterns”](#) in *ptg*
white space defaults : See [Section “White Space”](#) in *lex*
white space, in odin-expressions : See [Section “Products”](#) in *novice*
whitespace : See [Section “Value options”](#) in *clp*
window sizes : See [Section “X resources”](#) in *mon*
with : See [Section “Joined value options”](#) in *clp*
zero or more : See [Section “Star”](#) in *lex*
zero or one : See [Section “Query”](#) in *lex*