Interpreter

Generated by Doxygen 1.8.11

Contents

Chapter 1

Interpreter Main Page

1.1 Introduction

This is the introduction to iteration 3 of the interpreter project. So far we have created the scanner and parser for the interpreter. The scanner will read from a file and create a linked list of tokens that all contain Enumerated Tokentypes and using these Enumerated Tokentypes the parser is then able to generate an Abstract Syntax Tree (AST). The linked list of tokens is passed to the parser and using the Tokentypes is able to parse them into an AST and with each Node in the AST is able to unparse which will generate c++ code equivalent to the FCAL language we are interpreting from

1.1.1 Scanner

The scanner reads in characters from another file and and using regex expressions the scanner is able to categorize which characters are which Enumerated Tokentype. At the same time the scanner is also scanning for white space which it gets rid of using the regex for white space and bypasses the white space by moving the pointer reading the input file. After each character is properly categorized it is placed as a Token type in a linked last.

1.1.2 Parser

The Parser reads in the Token linked list from the scanner and goes through each Token in the linked list and generates a subclass according to the TokenType of each Token in the linked list. The first class generated is always the Root class which is the root of the AST that will be generate by the Parser. After this Root class has been generated other Stmt, Stmts, Expr, and Decl subclasses will be generated according to the TokenTypes of the rest of the Tokens in the Token linked list that was passed by the Scanner.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

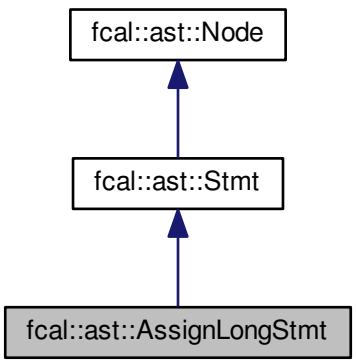
Class Documentation

4.1 fcal::ast::AssignLongStmt Class Reference

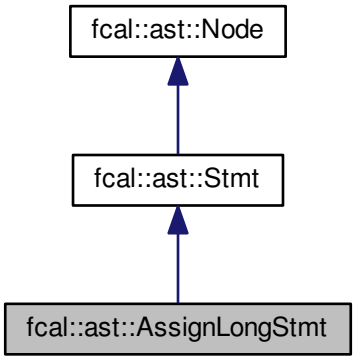
The AssignLongStmt class inherits directly from the abstract Stmt parent.

`#include <ast.h>`

Inheritance diagram for fcal::ast::AssignLongStmt:



Collaboration diagram for fcal::ast::AssignLongStmt:



Public Member Functions

- AssignLongStmt (VarName *var_name, Expr *expr, Expr *expr2, Expr *expr3)
- std::string unparse ()

  AssignLongStmt unparse() method.
- std::string **cpp_code** ()

4.1.1 Detailed Description

The AssignLongStmt class inherits directly from the abstract Stmt parent.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 fcal::ast::AssignLongStmt::AssignLongStmt (VarName * *var_name,* Expr * *expr,* Expr * *expr2,* Expr * *expr3*) [inline], [explicit]

AssignLongStmt production class takes the parameters: *var_name, *expr, expr2, and *expr3

Parameters

**var_name*	is the name of the variable being assigned
**expr*	is the first parameter in a matrix sequence
**expr2*	is the second parameter in a matrix sequence
**expr3*	is the expression being assigned to the specific matrix position

4.1.3 Member Function Documentation

4.1.3.1 std::string fcal::ast::AssignLongStmt::unparse ( ) [virtual]

AssignLongStmt unparse() method.

AssignLongStmt unparse() returns var_name_, expr_, expr2_ and expr3_.

Implements fcal::ast::Node.

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

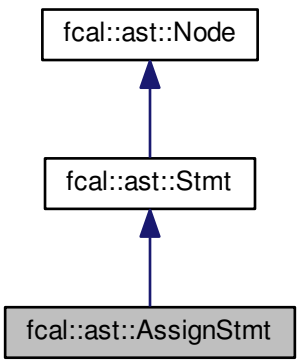
- include/ast.h
- src/ast.cc

4.2 fcal::ast::AssignStmt Class Reference

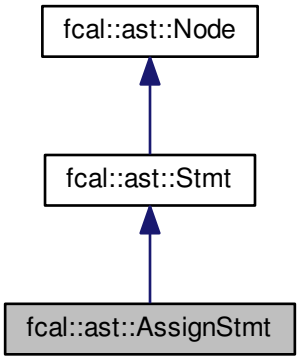
The AssignStmt class inherits directly from the abstract Stmt parent class.

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::AssignStmt:



Collaboration diagram for fcal::ast::AssignStmt:



Public Member Functions

- AssignStmt (VarName ∗var_name, Expr ∗expr)
- std::string unparse ()

  AssignStmt unparse() method.
- std::string **cpp_code** ()

4.2.1 Detailed Description

The AssignStmt class inherits directly from the abstract Stmt parent class.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 fcal::ast::AssignStmt::AssignStmt (VarName ∗ *var_name,* Expr ∗ *expr*) [inline], [explicit]

AssignStmt production class takes the parameters: ∗var_name and ∗expr

Parameters

∗*var_name*	is the name of the variable being assigned
∗*expr*	is the expression being assigned to the variable name

4.2.3 Member Function Documentation

4.2.3.1 std::string fcal::ast::AssignStmt::unparse ( ) [virtual]

AssignStmt unparse() method.

AssignStmt unparse() returns var_name_, expr_.

Implements fcal::ast::Node.

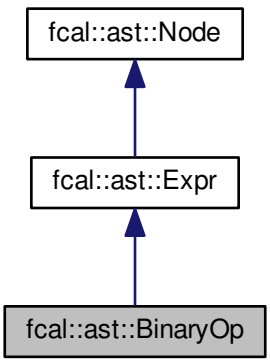
The documentation for this class was generated from the following files:

- include/ast.h
- src/ast.cc

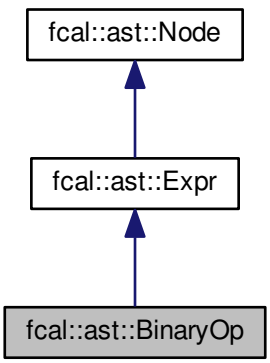
4.3 fcal::ast::BinaryOp Class Reference

`#include <ast.h>`

Inheritance diagram for fcal::ast::BinaryOp:



Collaboration diagram for fcal::ast::BinaryOp:



Public Member Functions

- BinaryOp (Expr *expr, std::string *op, Expr *expr2)
- std::string unparse ()

  BinaryOp unparse() method.
- std::string **cpp_code** ()

4.3.1 Detailed Description

The BinaryOp class inherits directly from the parent Expr class. The BinaryOp class combines the redundant nature of the implementing multiple production rule classes for the various binary operators including: *, /, +, -, >, >=, <, <=, ==, !=, && and ||.

The constructor determines the type of operator associated with expression by defining the *op to the lexeme of the prev_token_ for the matched signed.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 fcal::ast::BinaryOp::BinaryOp (**Expr** * ***expr,*** **std::string** * ***op,*** **Expr** * ***expr2***) [inline], [explicit]

BinaryOp production rules take the parameters: *expr, *op and *expr2

Parameters

**expr*	is the LHS expression
**op*	is the binary operator
**expr2*	is the RHS expression

4.3.3 Member Function Documentation

4.3.3.1 std::string fcal::ast::BinaryOp::unparse ( ) [virtual]

BinaryOp unparse() method.

BinaryOp returns the expr_, op_ and expr2_.

Implements fcal::ast::Node.

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

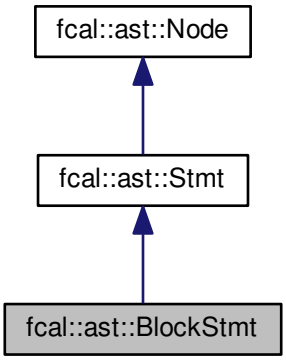
- include/ast.h
- src/ast.cc

4.4 fcal::ast::BlockStmt Class Reference

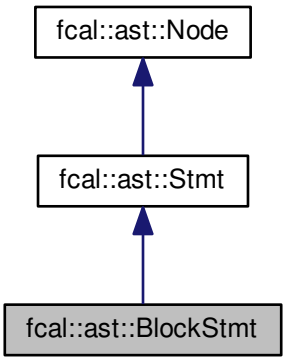
The BlockStmt class inherits directly from the abstract Stmt parent class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::BlockStmt:



Collaboration diagram for fcal::ast::BlockStmt:



Public Member Functions

- BlockStmt (Stmts *stmts)
- std::string unparse ()

  BlockStmt unparse() method.
- std::string **cpp_code** ()

4.4.1 Detailed Description

The BlockStmt class inherits directly from the abstract Stmt parent class.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 fcal::ast::BlockStmt::BlockStmt (Stmts * *stmts*) [inline], [explicit]

BlockStmt production class takes a single parameter: stmts

Parameters

**stmts*	statements

4.4.3 Member Function Documentation

4.4.3.1 std::string fcal::ast::BlockStmt::unparse ( ) [virtual]

BlockStmt unparse() method.

BlockStmt unparse() returns the stmts_ parameter.

Implements fcal::ast::Node.

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

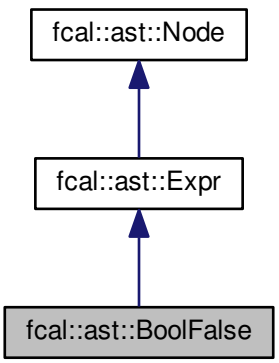
- include/ast.h
- src/ast.cc

4.5 fcal::ast::BoolFalse Class Reference

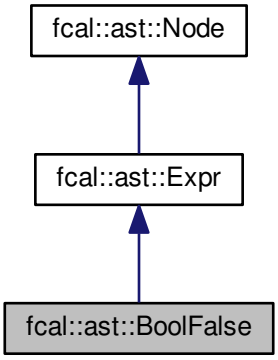
The BoolFalse class inherits directly from the abstract Expr class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::BoolFalse:



Collaboration diagram for fcal::ast::BoolFalse:



Public Member Functions

- BoolFalse ()
  BoolFalse() constructor.
- std::string unparse ()
  BoolFalse unparse() method.
- std::string **cpp_code** ()

4.5.1 Detailed Description

The BoolFalse class inherits directly from the abstract Expr class.

4.5.2 Member Function Documentation

4.5.2.1 std::string fcal::ast::BoolFalse::unparse ( ) [virtual]

BoolFalse unparse() method.

BoolFalse returns a "False" string for a boolean false.

Implements fcal::ast::Node.

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

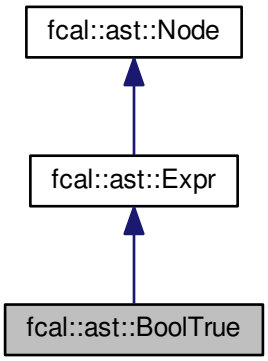
- include/ast.h
- src/ast.cc

4.6 fcal::ast::BoolTrue Class Reference

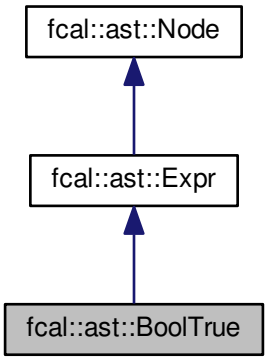
The BoolTrue class inherits directly from the abstract Expr class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::BoolTrue:



Collaboration diagram for fcal::ast::BoolTrue:



Public Member Functions

- BoolTrue ()

  BoolTrue() constructor.
- std::string unparse ()

  BoolTrue unparse() method.
- std::string **cpp_code** ()

4.6.1 Detailed Description

The BoolTrue class inherits directly from the abstract Expr class.

4.6.2 Member Function Documentation

4.6.2.1 std::string fcal::ast::BoolTrue::unparse ( ) [virtual]

BoolTrue unparse() method.

BoolTrue returns the "True" string for a boolean truth.

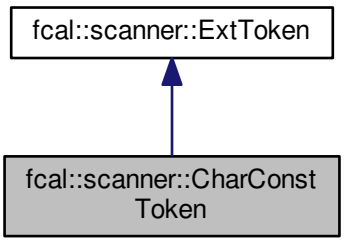
Implements fcal::ast::Node.

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

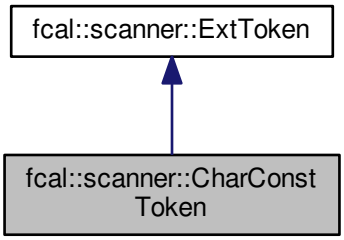
- include/ast.h
- src/ast.cc

4.7 fcal::scanner::CharConstToken Class Reference

Inheritance diagram for fcal::scanner::CharConstToken:



Collaboration diagram for fcal::scanner::CharConstToken:



Public Member Functions

- **CharConstToken** (parser::Parser ∗p, Token ∗t)
- parser::ParseResult **nud** ()
- std::string **description** ()

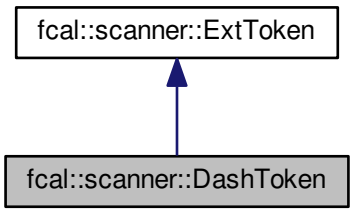
Additional Inherited Members

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

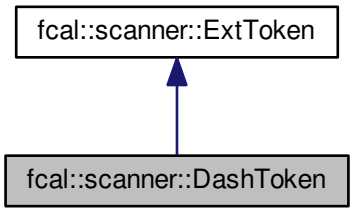
- include/ext_token.h

4.8 fcal::scanner::DashToken Class Reference

Inheritance diagram for fcal::scanner::DashToken:



Collaboration diagram for fcal::scanner::DashToken:



Public Member Functions

- **DashToken** (parser::Parser ∗p, Token ∗t)
- parser::ParseResult **led** (parser::ParseResult left)
- std::string **description** ()
- int **lbp** ()

Additional Inherited Members

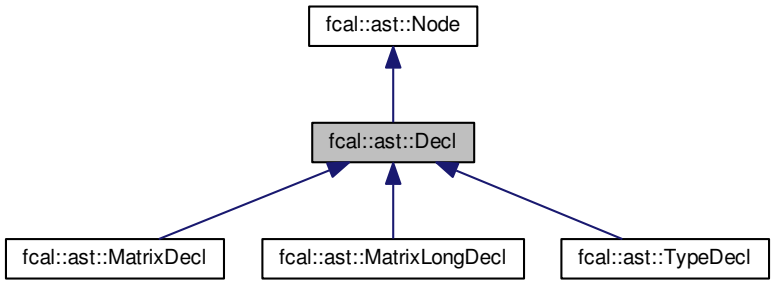
The documentation for this class was generated from the following file:

- include/ext_token.h

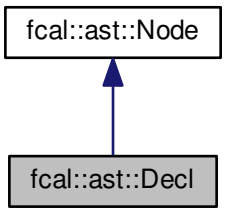
4.9 fcal::ast::Decl Class Reference

`#include <ast.h>`

Inheritance diagram for fcal::ast::Decl:



Collaboration diagram for fcal::ast::Decl:



Additional Inherited Members

4.9.1 Detailed Description

This is an abstract Decl class that inherits directly from the parent Node class.

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

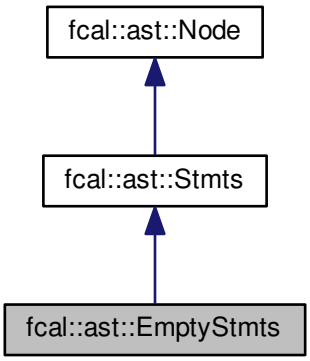
- include/ast.h

4.10 fcal::ast::EmptyStmts Class Reference

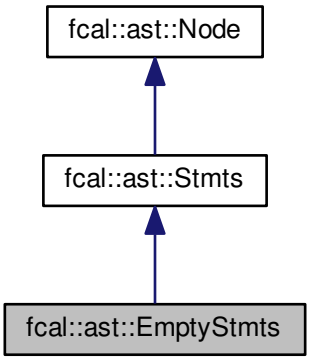
The EmptyStmts class inherits directly from the abstract Stmts parent class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::EmptyStmts:



Collaboration diagram for fcal::ast::EmptyStmts:



Public Member Functions

- EmptyStmts ()
  EmptyStmts Deconstructor.
- std::string unparse ()
  EmptyStmts unparse() method.
- std::string **cpp_code** ()

4.10.1 Detailed Description

The EmptyStmts class inherits directly from the abstract Stmts parent class.

4.10.2 Member Function Documentation

4.10.2.1 std::string fcal::ast::EmptyStmts::unparse ( ) [virtual]

EmptyStmts unparse() method.

EmptyStmts unparse() returns nothing.

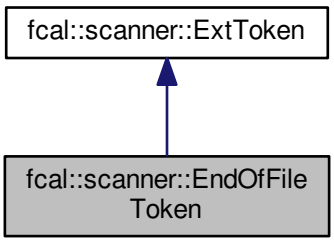
Implements fcal::ast::Node.

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

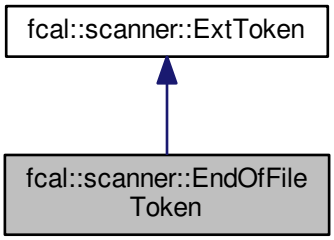
- include/ast.h
- src/ast.cc

4.11 fcal::scanner::EndOfFileToken Class Reference

Inheritance diagram for fcal::scanner::EndOfFileToken:



Collaboration diagram for fcal::scanner::EndOfFileToken:



Public Member Functions

- **EndOfFileToken** (parser::Parser *p, Token *t)
- std::string **description** ()

Additional Inherited Members

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

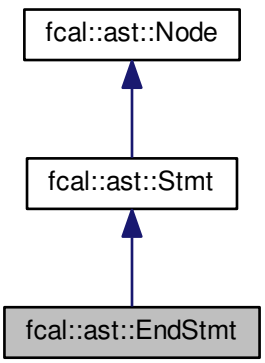
- include/ext_token.h

4.12 fcal::ast::EndStmt Class Reference

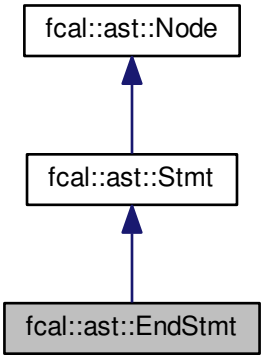
The EndStmt class inherits directly from the abstract Stmt parent class.

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::EndStmt:



Collaboration diagram for fcal::ast::EndStmt:



Public Member Functions

- EndStmt ()
  EndStmt() constructor.
- std::string unparse ()
  EndStmt unparse() method.
- std::string **cpp_code** ()

4.12.1 Detailed Description

The EndStmt class inherits directly from the abstract Stmt parent class.

4.12.2 Member Function Documentation

4.12.2.1 std::string fcal::ast::EndStmt::unparse ( ) [virtual]

EndStmt unparse() method.

EndStmt returns a semicolon (;) for end of line.

Implements fcal::ast::Node.

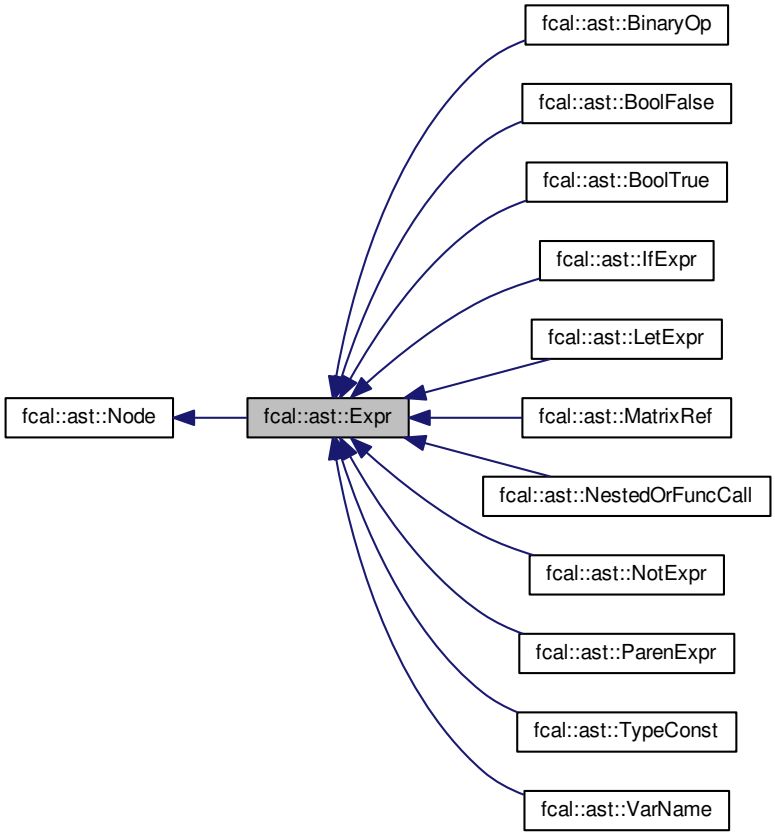
The documentation for this class was generated from the following files:

- include/ast.h
- src/ast.cc

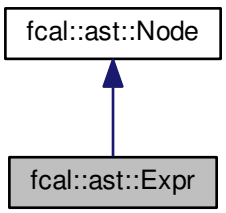
4.13 fcal::ast::Expr Class Reference

`#include <ast.h>`

Inheritance diagram for fcal::ast::Expr:



Collaboration diagram for fcal::ast::Expr:



Additional Inherited Members

4.13.1 Detailed Description

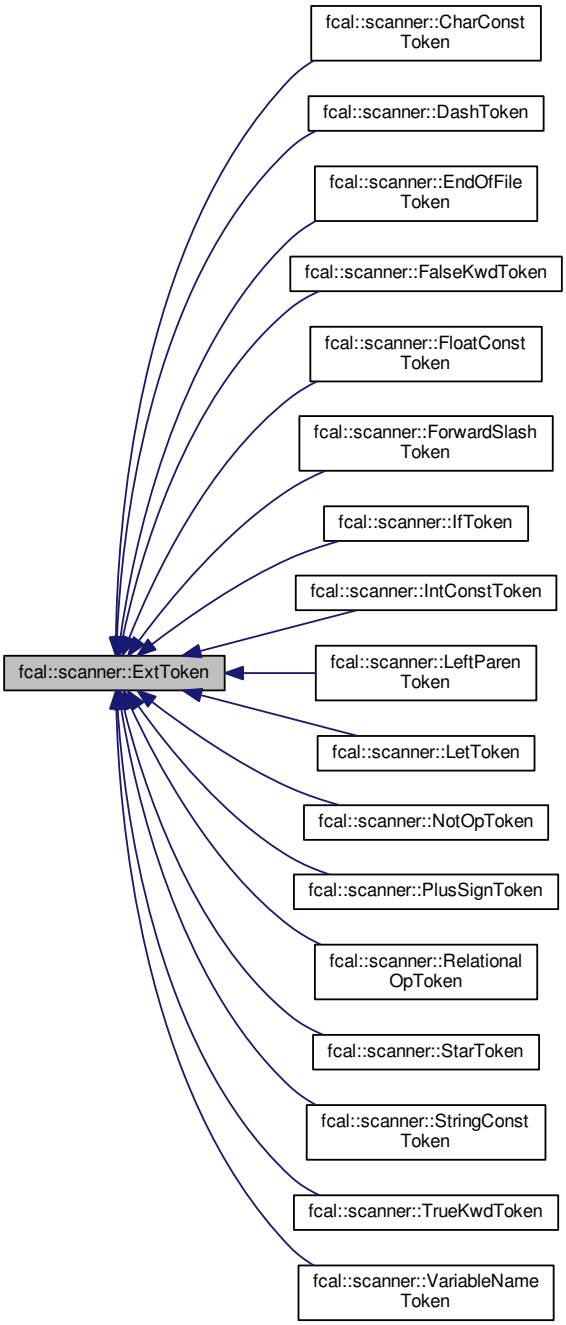
This is an abstract Expr class that inherits directly from the parent Node class.

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

- include/ast.h

4.14 fcal::scanner::ExtToken Class Reference

Inheritance diagram for fcal::scanner::ExtToken:



Public Member Functions

- **ExtToken** (parser::Parser ∗p, Token ∗t)
- **ExtToken** (parser::Parser ∗p, Token ∗t, std::string d)

- virtual parser::ParseResult **nud** (void)
- virtual parser::ParseResult **led** (parser::ParseResult left)
- ExtToken * **ExtendToken** (parser::Parser *p, Token *tokens)
- ExtToken * **ExtendTokenList** (parser::Parser *p, Token *tokens)
- virtual int **lbp** ()
- virtual std::string **description** ()
- std::string **lexeme** (void) const
- ExtToken * **next** (void) const
- scanner::TokenType **terminal** (void) const

Protected Member Functions

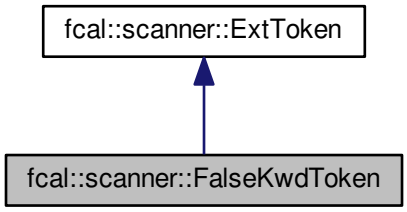
- parser::Parser * **parser** (void)

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

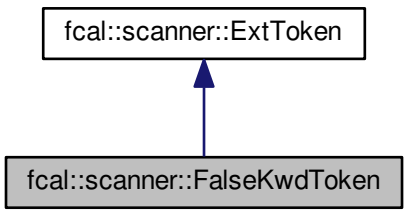
- include/ext_token.h
- src/ext_token.cc

4.15 fcal::scanner::FalseKwdToken Class Reference

Inheritance diagram for fcal::scanner::FalseKwdToken:



Collaboration diagram for fcal::scanner::FalseKwdToken:



Public Member Functions

- **FalseKwdToken** (parser::Parser *p, Token *t)
- parser::ParseResult **nud** ()
- std::string **description** ()

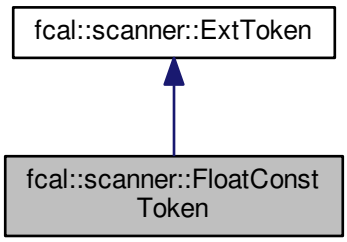
Additional Inherited Members

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

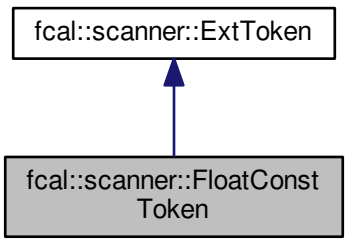
- include/ext_token.h

4.16 fcal::scanner::FloatConstToken Class Reference

Inheritance diagram for fcal::scanner::FloatConstToken:



Collaboration diagram for fcal::scanner::FloatConstToken:



Public Member Functions

- **FloatConstToken** (parser::Parser *p, Token *t)
- parser::ParseResult **nud** ()
- std::string **description** ()

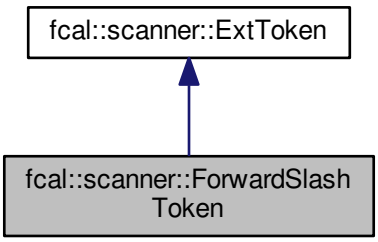
Additional Inherited Members

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

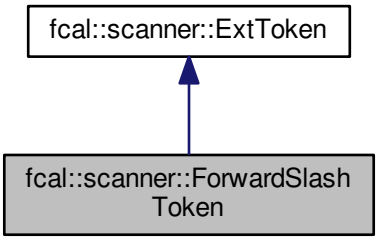
- include/ext_token.h

4.17 fcal::scanner::ForwardSlashToken Class Reference

Inheritance diagram for fcal::scanner::ForwardSlashToken:



Collaboration diagram for fcal::scanner::ForwardSlashToken:



Public Member Functions

- **ForwardSlashToken** (parser::Parser *p, Token *t)
- parser::ParseResult **led** (parser::ParseResult left)
- std::string **description** ()
- int **lbp** ()

Additional Inherited Members

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

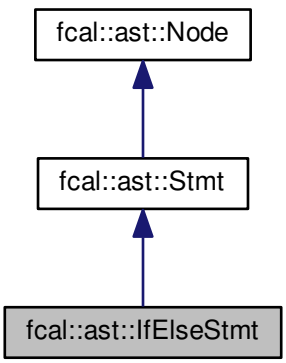
- include/ext_token.h

4.18 fcal::ast::IfElseStmt Class Reference

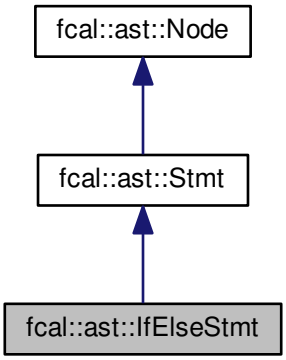
The IfElseStmt class inherits directly from the abstract Stmt parent class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::IfElseStmt:



Collaboration diagram for fcal::ast::IfElseStmt:



Public Member Functions

- IfElseStmt (Expr ∗expr, Stmt ∗stmt, Stmt ∗stmt2)
- std::string unparse ()

  IfElseStmt unparse method()
- std::string **cpp_code** ()

4.18.1 Detailed Description

The IfElseStmt class inherits directly from the abstract Stmt parent class.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 fcal::ast::IfElseStmt::IfElseStmt (Expr ∗ *expr,* Stmt ∗ *stmt,* Stmt ∗ *stmt2*) [inline],[explicit]

IfElseStmt production class takes the parameters: ∗expr, ∗stmt and ∗stmt2

Parameters

∗*expr*	expression of the if statement
∗*stmt*	statement of the then clause
∗*stmt2*	statement of the else clause

4.18.3 Member Function Documentation

4.18.3.1 std::string fcal::ast::IfElseStmt::unparse ( ) [virtual]

IfElseStmt unparse method()

IfElseStmt unparse() returns the expr_, stmt_ and stmt2_ parameters.

Implements fcal::ast::Node.

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

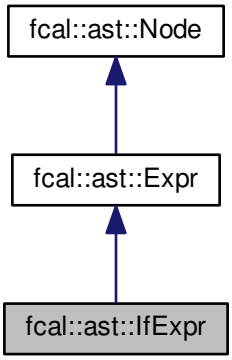
- include/ast.h
- src/ast.cc

4.19 fcal::ast::IfExpr Class Reference

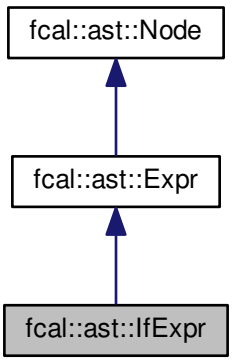
The IfExpr class inherits directly from the abstract Expr class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::IfExpr:



Collaboration diagram for fcal::ast::IfExpr:



Public Member Functions

- IfExpr (Expr *expr, Expr *expr2, Expr *expr3)
- std::string unparse ()

  IfExpr unparse() method.
- std::string **cpp_code** ()

4.19.1 Detailed Description

The IfExpr class inherits directly from the abstract Expr class.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 fcal::ast::IfExpr::IfExpr (Expr * *expr,* Expr * *expr2,* Expr * *expr3*) [inline],[explicit]

IfExpr production rules take the paramters: *expr, *expr2 and *expr3

Parameters

*expr	is the if expression
*expr2	is the then expression
*expr3	is the else expression

4.19.3 Member Function Documentation

4.19.3.1 std::string fcal::ast::IfExpr::unparse ( ) [virtual]

IfExpr unparse() method.

IfExpr returns expr_, expr2_ and expr3_ paramters.

Implements fcal::ast::Node.

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

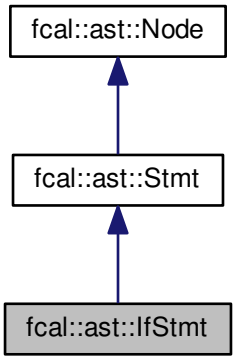
- include/ast.h
- src/ast.cc

4.20 fcal::ast::IfStmt Class Reference

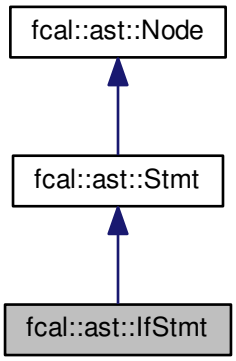
The IfStmt class inherits directly from the abstract Stmt parent class.

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::IfStmt:



Collaboration diagram for fcal::ast::IfStmt:



Public Member Functions

- IfStmt (Expr *expr, Stmt *stmt)
- std::string unparse ()

  IfStmt unparse() method.
- std::string **cpp_code** ()

4.20.1 Detailed Description

The IfStmt class inherits directly from the abstract Stmt parent class.

4.20.2 Constructor & Destructor Documentation

4.20.2.1 fcal::ast::IfStmt::IfStmt (Expr * *expr*, Stmt * *stmt*) [inline], [explicit]

IfStmt production class takes the parameters: *expr and *stmt

Parameters

**expr*	expression of the if statement
**stmt*	statement of the then clause

4.20.3 Member Function Documentation

4.20.3.1 std::string fcal::ast::IfStmt::unparse ( ) [virtual]

IfStmt unparse() method.

IfStmt unparse() returns the expr_ and stmt_ parameters.

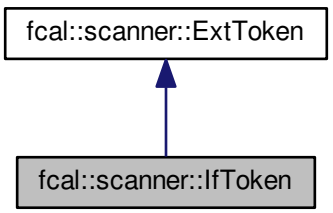
Implements fcal::ast::Node.

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

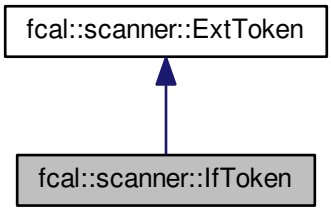
- include/ast.h
- src/ast.cc

4.21 fcal::scanner::IfToken Class Reference

Inheritance diagram for fcal::scanner::IfToken:



Collaboration diagram for fcal::scanner::IfToken:



Public Member Functions

- **IfToken** (parser::Parser *p, Token *t)
- parser::ParseResult **nud** ()
- std::string **description** ()
- int **lbp** ()

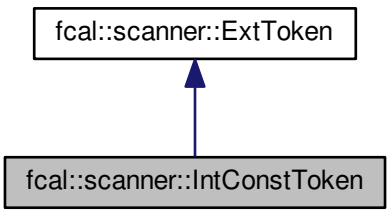
Additional Inherited Members

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

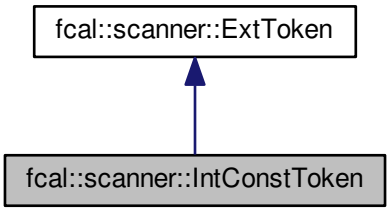
- include/ext_token.h

4.22 fcal::scanner::IntConstToken Class Reference

Inheritance diagram for fcal::scanner::IntConstToken:



Collaboration diagram for fcal::scanner::IntConstToken:



Public Member Functions

- **IntConstToken** (parser::Parser *p, Token *t)
- parser::ParseResult **nud** ()
- std::string **description** ()

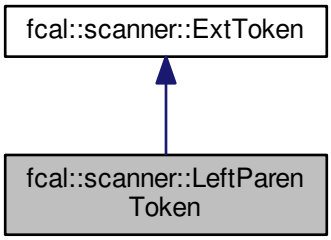
Additional Inherited Members

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

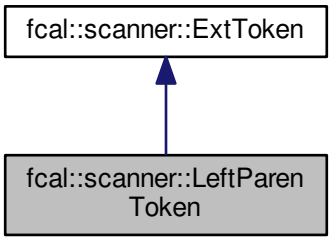
- include/ext_token.h

4.23 fcal::scanner::LeftParenToken Class Reference

Inheritance diagram for fcal::scanner::LeftParenToken:



Collaboration diagram for fcal::scanner::LeftParenToken:



Public Member Functions

- **LeftParenToken** (parser::Parser *p, Token *t)
- parser::ParseResult **nud** ()
- std::string **description** ()
- int **lbp** ()

Additional Inherited Members

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

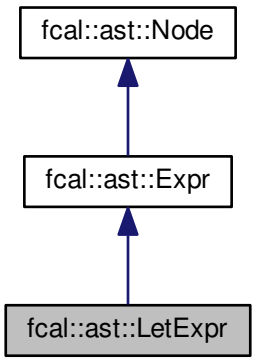
- include/ext_token.h

4.24 fcal::ast::LetExpr Class Reference

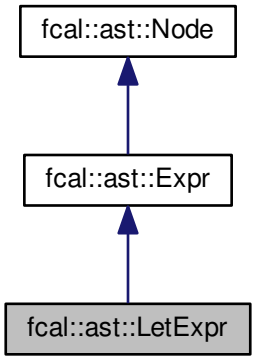
The LetExpr class inherits directly from the abstract Expr class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::LetExpr:



Collaboration diagram for fcal::ast::LetExpr:



Public Member Functions

- LetExpr (Stmts ∗stmts, Expr ∗expr)
- std::string unparse ()

  LetExpr unparse() method.
- std::string **cpp_code** ()

4.24.1 Detailed Description

The LetExpr class inherits directly from the abstract Expr class.

4.24.2 Constructor & Destructor Documentation

4.24.2.1 fcal::ast::LetExpr::LetExpr (Stmts ∗ *stmts*, Expr ∗ *expr*) [inline], [explicit]

LetExpr production rules take the parameters: ∗stmts and ∗expr

Parameters

∗*stmts*	are the statements between let and in
∗*expr*	is the expression after in and before end

4.24.3 Member Function Documentation

4.24.3.1 std::string fcal::ast::LetExpr::unparse ( ) [virtual]

LetExpr unparse() method.

LetExpr returns the stmts_ and expr_ paramters.

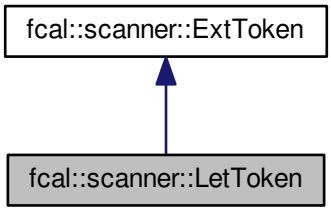
Implements fcal::ast::Node.

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

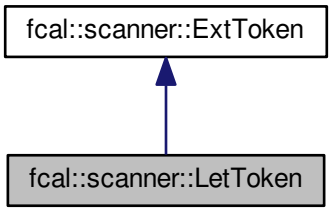
- include/ast.h
- src/ast.cc

4.25 fcal::scanner::LetToken Class Reference

Inheritance diagram for fcal::scanner::LetToken:



Collaboration diagram for fcal::scanner::LetToken:



Public Member Functions

- **LetToken** (parser::Parser *p, Token *t)
- parser::ParseResult **nud** ()
- std::string **description** ()
- int **lbp** ()

Additional Inherited Members

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

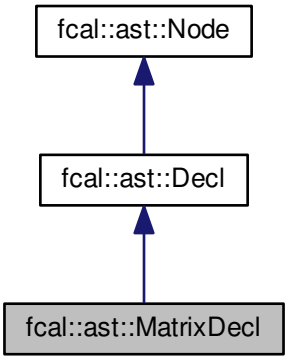
- include/ext_token.h

4.26 fcal::ast::MatrixDecl Class Reference

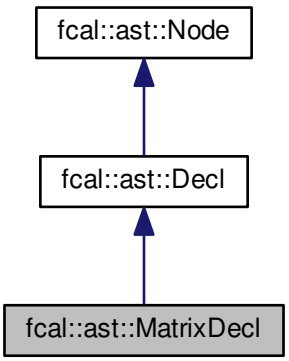
The MatrixDecl class inherits directly from the abstract Decl class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::MatrixDecl:



Collaboration diagram for fcal::ast::MatrixDecl:



Public Member Functions

- MatrixDecl (VarName *var_name, Expr *expr)
- std::string unparse ()

  MatrixDecl unparse() method.
- std::string **cpp_code** ()

4.26.1 Detailed Description

The MatrixDecl class inherits directly from the abstract Decl class.

4.26.2 Constructor & Destructor Documentation

4.26.2.1 fcal::ast::MatrixDecl::MatrixDecl (VarName * *var_name*, Expr * *expr*) [inline], [explicit]

MatrixDecl production class takes the parameters: *var_name and *expr

Parameters

**var_name*	is the name of the variable being assigned
**expr*	is the expression being assigned to the variable

4.26.3 Member Function Documentation

4.26.3.1 std::string fcal::ast::MatrixDecl::unparse ( ) [virtual]

MatrixDecl unparse() method.

MatrixDecl returns var_name_ and expr_.

Implements fcal::ast::Node.

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

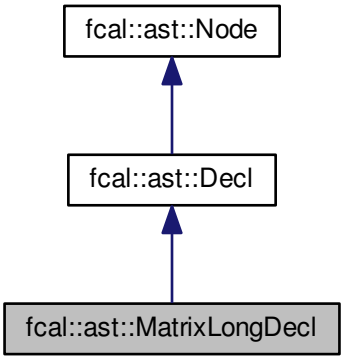
- include/ast.h
- src/ast.cc

4.27 fcal::ast::MatrixLongDecl Class Reference

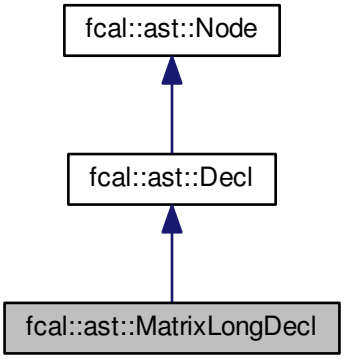
The MatrixLongDecl class inherits directly from the abstract Decl class.

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::MatrixLongDecl:



Collaboration diagram for fcal::ast::MatrixLongDecl:



Public Member Functions

- MatrixLongDecl (VarName *var_name, Expr *expr, Expr *expr2, VarName *var_name2, VarName *var_name3, Expr *expr3)
- std::string unparse ()

  MatrixLongDecl unparse() method.
- std::string **cpp_code** ()

4.27.1 Detailed Description

The MatrixLongDecl class inherits directly from the abstract Decl class.

4.27.2 Constructor & Destructor Documentation

4.27.2.1 fcal::ast::MatrixLongDecl::MatrixLongDecl (VarName * *var_name,* Expr * *expr,* Expr * *expr2,* VarName * *var_name2,* VarName * *var_name3,* Expr * *expr3*) [inline], [explicit]

MatrixLogDecl production class takes the parameters: *var_name, *expr, expr2, *var_name2, *var_name3, and *expr3

Parameters

**var_name*	names the variable referencing the matrix
**expr*	first parameter of the matrix
**expr2*	second parameter of the matrix
**var_name2*	first variable reference
**var_name3*	secondary variable reference
**expr3*	expression being assigned

4.27.3 Member Function Documentation

4.27.3.1 std::string fcal::ast::MatrixLongDecl::unparse ( ) [virtual]

MatrixLongDecl unparse() method.

MatrixLongDecl returns var_name_, expr_, expr2_, var_name2_, var_name3_, and expr3_

Implements fcal::ast::Node.

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

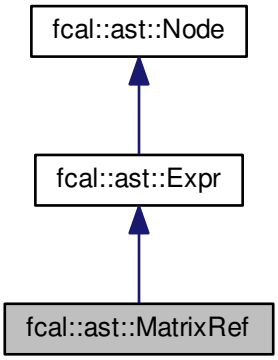
- include/ast.h
- src/ast.cc

4.28 fcal::ast::MatrixRef Class Reference

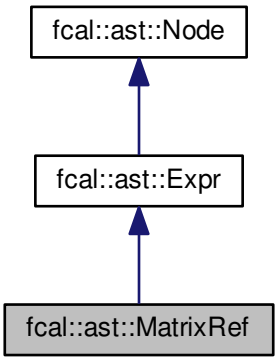
The MatrixRef class inherits directly from the abstract Expr class.

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::MatrixRef:



Collaboration diagram for fcal::ast::MatrixRef:



Public Member Functions

- MatrixRef (VarName *var_name, Expr *expr, Expr *expr2)
- std::string unparse ()

  MatrixRef unparse() method.
- std::string **cpp_code** ()

4.28.1 Detailed Description

The MatrixRef class inherits directly from the abstract Expr class.

4.28.2 Constructor & Destructor Documentation

4.28.2.1 fcal::ast::MatrixRef::MatrixRef (VarName * *var_name,* Expr * *expr,* Expr * *expr2*) [inline], [explicit]

MatrixRef production rules take the parameters: *var_name, *expr, and expr2

Parameters

**var_name*	is the name of the matrix reference
**expr*	is the first parameter
**expr2*	is the second parameter

4.28.3 Member Function Documentation

4.28.3.1 std::string fcal::ast::MatrixRef::unparse ( ) [virtual]

MatrixRef unparse() method.

MatrixRef returns the var_name_, expr_ and expr2_ parameters.

Implements fcal::ast::Node.

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

- include/ast.h
- src/ast.cc

4.29 MySequence< T, N > Class Template Reference

Public Member Functions

- void **set_member** (int x, T value)
- T **get_member** (int x)

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

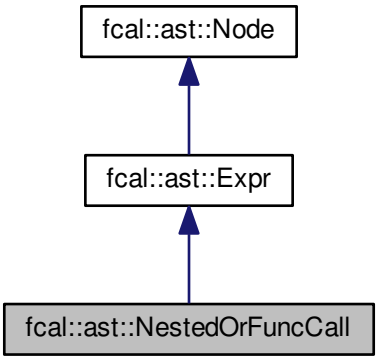
- src/templates.cc

4.30 fcal::ast::NestedOrFuncCall Class Reference

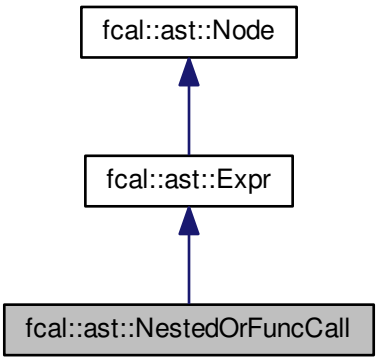
The NestedOrFuncCall class inherits directly from the abstract Expr class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::NestedOrFuncCall:



Collaboration diagram for fcal::ast::NestedOrFuncCall:



Public Member Functions

- NestedOrFuncCall (VarName *var_name, Expr *expr)
- std::string unparse ()

  NestedOrFuncCall unparse() method.
- std::string **cpp_code** ()

4.30.1 Detailed Description

The NestedOrFuncCall class inherits directly from the abstract Expr class.

4.30.2 Constructor & Destructor Documentation

4.30.2.1 fcal::ast::NestedOrFuncCall::NestedOrFuncCall (VarName * *var_name,* Expr * *expr*) [inline], [explicit]

NestedOrFuncCall production rules take the parameters: *var_name and *expr

Parameters

**var_name*	is the variable name
**expr*	is the nested expression

4.30.3 Member Function Documentation

4.30.3.1 std::string fcal::ast::NestedOrFuncCall::unparse ( ) [virtual]

NestedOrFuncCall unparse() method.

NestedOrFuncCall returns the var_name_ and expr_ paramters.

Implements fcal::ast::Node.

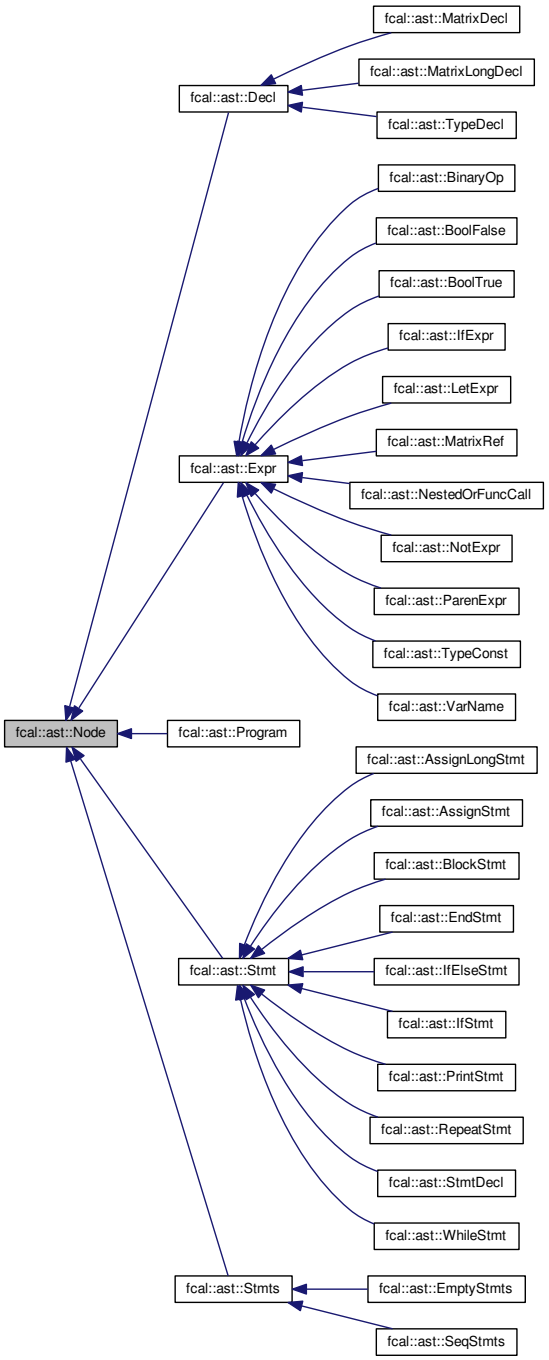
The documentation for this class was generated from the following files:

- include/ast.h
- src/ast.cc

4.31 fcal::ast::Node Class Reference

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::Node:



Public Member Functions

- virtual std::string unparse ()=0

  virtual unparse() method
- virtual ~Node ()

  Node() deconstructor.

4.31.1 Detailed Description

The abstract Node base class is the parent to all classes within the production rules. All further classes will inherit the unparse() function.

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

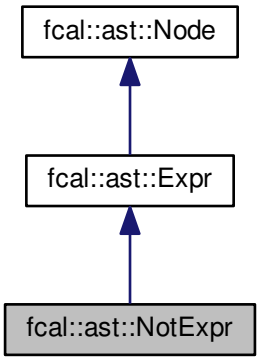
- include/ast.h

4.32 fcal::ast::NotExpr Class Reference

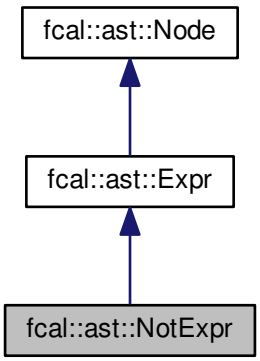
The NotExpr class inherits directly from the abstract Expr class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::NotExpr:



Collaboration diagram for fcal::ast::NotExpr:



Public Member Functions

- NotExpr (Expr *expr)
- std::string unparse ()

  NotExpr unparse() method.
- std::string **cpp_code** ()

4.32.1 Detailed Description

The NotExpr class inherits directly from the abstract Expr class.

4.32.2 Constructor & Destructor Documentation

4.32.2.1 fcal::ast::NotExpr::NotExpr (Expr * *expr*) [inline], [explicit]

NotExpr production rules take the parameter: *expr

Parameters

**expr*	is the expression being negated

4.32.3 Member Function Documentation

4.32.3.1 std::string fcal::ast::NotExpr::unparse ( ) [virtual]

NotExpr unparse() method.

NotExpr returns a negated expr_ parameter.

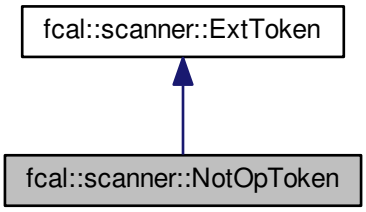
Implements fcal::ast::Node.

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

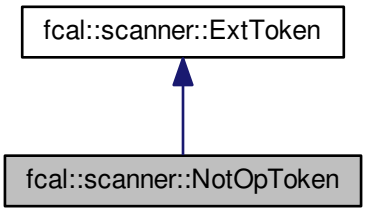
- include/ast.h
- src/ast.cc

4.33 fcal::scanner::NotOpToken Class Reference

Inheritance diagram for fcal::scanner::NotOpToken:



Collaboration diagram for fcal::scanner::NotOpToken:



Public Member Functions

- **NotOpToken** (parser::Parser ∗p, Token ∗t)
- parser::ParseResult **nud** ()
- std::string **description** ()

Additional Inherited Members

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

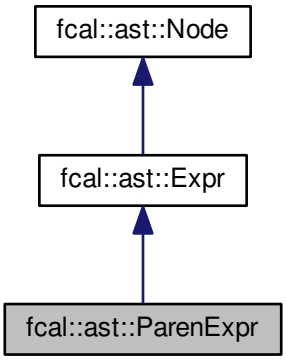
- include/ext_token.h

4.34 fcal::ast::ParenExpr Class Reference

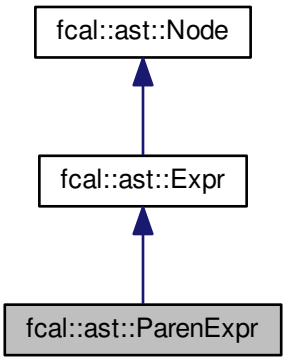
The ParenExpr class inherits directly from the abstract Expr class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::ParenExpr:



Collaboration diagram for fcal::ast::ParenExpr:



Public Member Functions

- ParenExpr (Expr *expr)
- std::string unparse ()

  ParenExpr unparse() method.
- std::string **cpp_code** ()

4.34.1 Detailed Description

The ParenExpr class inherits directly from the abstract Expr class.

4.34.2 Constructor & Destructor Documentation

4.34.2.1 fcal::ast::ParenExpr::ParenExpr (Expr * *expr*) [inline],[explicit]

ParenExpr production rules take a single parameter: *expr

Parameters

**expr*	is expression nested between parantheses

4.34.3 Member Function Documentation

4.34.3.1 std::string fcal::ast::ParenExpr::unparse ( ) [virtual]

ParenExpr unparse() method.

ParenExpr returns the expr_ paramter.

Implements fcal::ast::Node.

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

- include/ast.h
- src/ast.cc

4.35 fcal::parser::Parser Class Reference

Public Member Functions

- ~Parser (void)

  Parser deconstructor function.
- ParseResult Parse (const char *text)

  Parse constructor function.
- ParseResult ParseProgram ()

  ParseProgram creates the first node in the AST, the Program Node.
- ParseResult parse_decl ()
- ParseResult parse_standard_decl ()
- ParseResult parse_matrix_decl ()
- ParseResult parse_stmts ()
- ParseResult parse_stmt ()
- ParseResult **parse_expr** (int rbp)
- ParseResult parse_true_kwd ()

- ParseResult parse_false_kwd ()
- ParseResult parse_int_const ()
- ParseResult parse_float_const ()
- ParseResult parse_string_const ()
- ParseResult **parse_char_const** ()
- ParseResult parse_variable_name ()
- ParseResult parse_nested_expr ()
- ParseResult parse_not_expr ()
- ParseResult parse_let_expr ()
- ParseResult parse_if_expr ()
- ParseResult parse_addition (ParseResult left)
- ParseResult parse_multiplication (ParseResult left)
- ParseResult parse_subtraction (ParseResult left)
- ParseResult parse_division (ParseResult left)
- ParseResult parse_relational_expr (ParseResult left)
- void **match** (const scanner::TokenType &tt)
- bool **attempt_match** (const scanner::TokenType &tt)
- bool **next_is** (const scanner::TokenType &tt)
- void **next_token** (void)

4.35.1 Member Function Documentation

4.35.1.1 ParseResult fcal::parser::Parser::parse_addition (ParseResult *prLeft*)

parse_addition will generate a BinaryOp with parameters expr, "+", and expr2

4.35.1.2 ParseResult fcal::parser::Parser::parse_decl ( )

parse_decl will categorize what type of declaration using the current Token types to either parse_matrix_decl or parse_standard_decl

4.35.1.3 ParseResult fcal::parser::Parser::parse_division (ParseResult *prLeft*)

parse_division will generate a BinaryOp with parameters expr, "/", and expr2

4.35.1.4 ParseResult fcal::parser::Parser::parse_false_kwd ( )

parser_false_kwd idenitifies the current node's Token type and if it is kFalseKwd then it generates a BoolFalse subclass

4.35.1.5 ParseResult fcal::parser::Parser::parse_float_const ( )

parse_float_const identifies the current node's Token type and if it is kFloatConst then it generates a TypeConst subclass and passes in a "float" lexeme with it

4.35.1.6 ParseResult fcal::parser::Parser::parse_if_expr ( )

parse_if_expr will generate an IfExpr subclass with parameters expr, expr2, and expr3

4.35.1.7 ParseResult fcal::parser::Parser::parse_int_const ( )

parse_int_const identifies the current node's Token type and if it is kIntConst then it generates a TypeConst subclass and passes in a "int" lexeme with it

4.35.1.8 ParseResult fcal::parser::Parser::parse_let_expr ( )

parse_let_expr will generate a LetExpr with parameters stmts and expr

4.35.1.9 ParseResult fcal::parser::Parser::parse_matrix_decl ( )

parse_matrix_decl parses a matrix declaration. If the second token is a left square bracker then it will parse according to the MatrixLongDecl, but there is not left square bracket then it will parse according to the regular MatrixDecl

4.35.1.10 ParseResult fcal::parser::Parser::parse_multiplication (ParseResult *prLeft*)

parse_multiplication will generate a BinaryOp with parameters expr, "∗", and expr2

4.35.1.11 ParseResult fcal::parser::Parser::parse_nested_expr ( )

parse_nested_expr will generate a ParenExpr subclass with parameter expr

4.35.1.12 ParseResult fcal::parser::Parser::parse_not_expr ( )

parse_not_expr will generate a NotExpr with parameter expr

4.35.1.13 ParseResult fcal::parser::Parser::parse_relational_expr (ParseResult *prLeft*)

parse_relational_expr will generate a BinaryOp with expr, whichever relational expression, and expr2

4.35.1.14 ParseResult fcal::parser::Parser::parse_standard_decl ( )

parse_standard_decl parses a type declaration. The decl_type will be passed to the general TypeDecl subclass and the decl_type will be placed in front of the varName ensuring correct parsing

4.35.1.15 ParseResult fcal::parser::Parser::parse_stmt ( )

parse_stmt will categorize the type of statement by identifying the keyword and will create the according subclass for it. If the current token is a keyword associated with declarations; kIntKwd, kFloatKwd, etc. it will create a StmtDecl subclass. If the current token is the keyword kLeftCurly then a BlockStmt subclass will be created If the current token is the keyword kIfKwd then a IfStmt subclass will be created, but if there is a token after that is kElseKwd then the subclass IfElseStmt will created instead If the current token is the keyword kVariableName and the next token is of type kLeftSquare then a AssignLongStmt subclass will be created, but if then tokens are just kVariableName and kAssign then an AssignStmt will be created If the current token is the keyword kPrintKwd then a PrintStmt will be created If the current token is the keyword kRepeatKwd then a RepeatStmt will be created If the current token is the keyword kWhileKwd then a WhileStmt will be created If the current token is the keyword kSemiColon then an EndStmt will be created If there is current token then throw an error message

4.35.1.16 ParseResult fcal::parser::Parser::parse_stmts ( )

parse_stmts will parse EmptyStmts if it is the last Node of the AST, but if the next Node in the AST is neither a kRightCurly or a kInKwd then it will continue parsing with SeqStmts

4.35.1.17 ParseResult fcal::parser::Parser::parse_string_const ( )

parse_string_const identifies the current node's Token type and if it is kStringConst then it generates a TypeConst subclass and passes in a "string" lexeme with it

4.35.1.18 ParseResult fcal::parser::Parser::parse_subtraction (ParseResult *prLeft*)

parse_subtraction will generate a BinaryOp with parameters expr, "-", and expr2

4.35.1.19 ParseResult fcal::parser::Parser::parse_true_kwd ( )

parse_true_kwd identifies the current node's Token type and if it is kTrueKwd then it generates a BoolTrue subclass

4.35.1.20 ParseResult fcal::parser::Parser::parse_variable_name ( )

parse_variable_name identifies the current token's type and creates a subclass according to it If the current token is the keyword kLeftSquare then a MatrixRef subclass will be created If the current token is the keyword kLeftParen then a NestedOrFuncCall will be created Else if the current token matches none of these then it creates a VarName subclass

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

- include/parser.h
- src/parser.cc

4.36 fcal::parser::ParseResult Class Reference

Public Member Functions

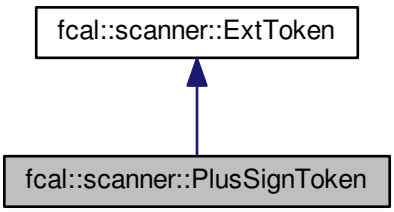
- bool **ok** (void) const
- void **ok** (bool result_in)
- std::string **errors** (void) const
- void **errors** (const std::string str_in)
- ast::Node * **ast** (void)
- void **ast** (ast::Node *Node_ptr)

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

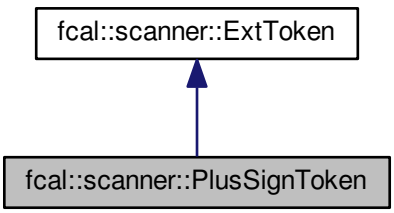
- include/parse_result.h

4.37 fcal::scanner::PlusSignToken Class Reference

Inheritance diagram for fcal::scanner::PlusSignToken:



Collaboration diagram for fcal::scanner::PlusSignToken:



Public Member Functions

- **PlusSignToken** (parser::Parser *p, Token *t)
- parser::ParseResult **led** (parser::ParseResult left)
- std::string **description** ()
- int **lbp** ()

Additional Inherited Members

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

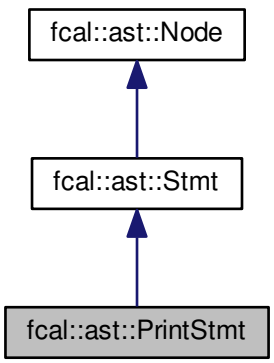
- include/ext_token.h

4.38 fcal::ast::PrintStmt Class Reference

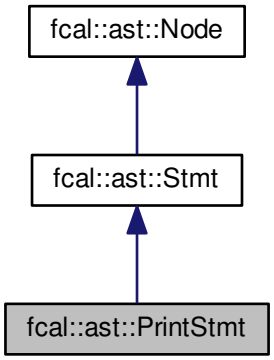
The PrintStmt class inherits directly from the abstract Stmt parent class.

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::PrintStmt:



Collaboration diagram for fcal::ast::PrintStmt:



Public Member Functions

- PrintStmt (Expr *expr)
- std::string unparse ()

  PrintStmt unparse() method.
- std::string **cpp_code** ()

4.38.1 Detailed Description

The PrintStmt class inherits directly from the abstract Stmt parent class.

4.38.2 Constructor & Destructor Documentation

4.38.2.1 fcal::ast::PrintStmt::PrintStmt (Expr * *expr*) [inline], [explicit]

PrintStmt production class takes a single parameter: *expr

Parameters

*expr	parameter of the printing expression

4.38.3 Member Function Documentation

4.38.3.1 std::string fcal::ast::PrintStmt::unparse ( ) [virtual]

PrintStmt unparse() method.

PrintStmt unparse() returns the expr_ parameter.

Implements fcal::ast::Node.

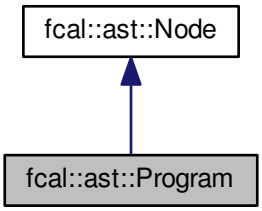
The documentation for this class was generated from the following files:

- include/ast.h
- src/ast.cc

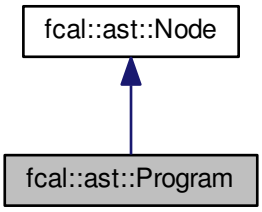
4.39 fcal::ast::Program Class Reference

`#include <ast.h>`

Inheritance diagram for fcal::ast::Program:



Collaboration diagram for fcal::ast::Program:



Public Member Functions

- Program (VarName *v, Stmts *s)
- std::string unparse ()

  Program unparse() method.
- std::string **cpp_code** ()
- virtual ~Program ()

  Program() deconstructor.

4.39.1 Detailed Description

The Program class, otherwise known as the Root class, inherits directly from the abstract parent Node class. The Program class starts the production rules to build the AST.

4.39.2 Constructor & Destructor Documentation

4.39.2.1 fcal::ast::Program::Program (VarName * v, Stmts * s) [inline], [explicit]

Program production class takes two parameters: *v and *s

Parameters

*v	the name of the program
*s	statements on the RHS of the tree

4.39.2.2 fcal::ast::Program::~Program ( ) [virtual]

Program() deconstructor.

Program deconstructor method.

4.39.3 Member Function Documentation

4.39.3.1 std::string fcal::ast::Program::unparse ( ) [virtual]

Program unparse() method.

Program unparse() returns the var_name_ and stmts_ parameters.

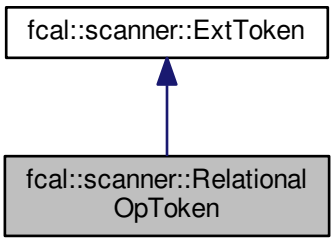
Implements fcal::ast::Node.

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

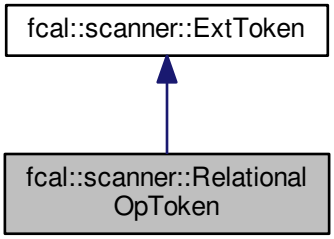
- include/ast.h
- src/ast.cc

4.40 fcal::scanner::RelationalOpToken Class Reference

Inheritance diagram for fcal::scanner::RelationalOpToken:



Collaboration diagram for fcal::scanner::RelationalOpToken:



Public Member Functions

- **RelationalOpToken** (parser::Parser ∗p, Token ∗t, std::string d)
- parser::ParseResult **led** (parser::ParseResult left)
- int **lbp** ()

Additional Inherited Members

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

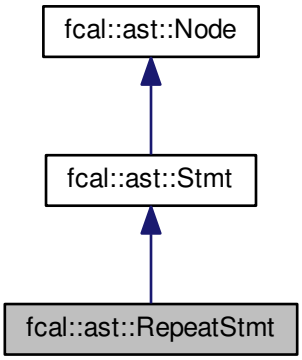
- include/ext_token.h

4.41 fcal::ast::RepeatStmt Class Reference

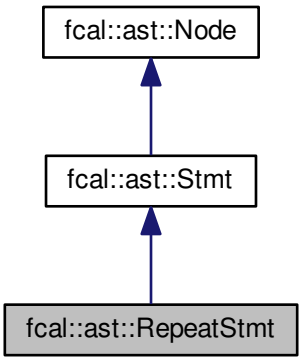
The RepeatStmt class inherits directly from the abstract Stmt parent class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::RepeatStmt:



Collaboration diagram for fcal::ast::RepeatStmt:



Public Member Functions

- RepeatStmt (VarName *var_name, Expr *expr, Expr *expr2, Stmt *stmt)
- std::string unparse ()

  RepeatStmt unparse() method.
- std::string **cpp_code** ()

4.41.1 Detailed Description

The RepeatStmt class inherits directly from the abstract Stmt parent class.

4.41.2 Constructor & Destructor Documentation

4.41.2.1 fcal::ast::RepeatStmt::RepeatStmt (VarName * *var_name,* Expr * *expr,* Expr * *expr2,* Stmt * *stmt*) [inline],[explicit]

RepeatStmt production class takes the parameters: *var_name, *expr, expr2, and *stmt

Parameters

**var_name*	is the name of the variable being assigned
**expr*	is the start parameter
**expr2*	is the end parameter
**stmt*	is the statement being repeated

4.41.3 Member Function Documentation

4.41.3.1 std::string fcal::ast::RepeatStmt::unparse ( ) [virtual]

RepeatStmt unparse() method.

RepeatStmt returns var_name_, expr_, expr2_ and stmt_.

Implements fcal::ast::Node.

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

- include/ast.h
- src/ast.cc

4.42 fcal::scanner::Scanner Class Reference

`#include <scanner.h>`

Public Member Functions

- Scanner ()
  Scanner() constructor.
- ~Scanner ()
  Scanner() deconstructor.
- void InitRegexTokenArray ()
  Initalizes an array of regular expressions associated with its token type.
- Token * Scan (const char *text)
  Scan muethod that reads through a file to determine token type matches.
- int consume_whitespace_and_comments (regex_t *white_space, regex_t *block_comment, regex_t *single_comment, const char *text)

Public Attributes

- regex_t * **regex_token_array** [45]

4.42.1 Detailed Description

The Scanner class defines the various methods and attributes associated with scanning in a file to determine the token types.

4.42.2 Constructor & Destructor Documentation

4.42.2.1 fcal::scanner::Scanner::Scanner ( )

Scanner() constructor.

Scanner() constructor; calls to InitRegexTokenArray to initalize array.

4.42.3 Member Function Documentation

4.42.3.1 int fcal::scanner::Scanner::consume_whitespace_and_comments (regex_t * *white_space,* regex_t * *block_comment,* regex_t * *single_comment,* const char * *text*)

The consume_whitespace_and_comments method scans through a file or string and looks for comment lines, block comments and white spaces and removes them from consumption so that the Scan method can pass through the file without mistaking one of them for a token type.

4.42.3.2 void fcal::scanner::Scanner::InitRegexTokenArray ( )

Initalizes an array of regular expressions associated with its token type.

InitRegexTokenArray creates an array of regular expressions that matches to the token types, and the array is indexed based on the enum kTokenEnumType variable names.

4.42.3.3 Token * fcal::scanner::Scanner::Scan (const char * *text*)

Scan muethod that reads through a file to determine token type matches.

The Scan method reads in a file or string, but if the file or string is determined to be NULL, than the Scan method will return NULL and terminate. If the Scan continues then it will check for an EOF character before continuing to scan in all the characters. As the Scan method scans the file, it will iterate through the InitRegexTokenArray to determine possible matches to the string; as it finds a given match, it stores the longest max_num_matched_chars and once it's done iterating through the string, it will return the token match_type. As it returns the match_type it will push the determined string of characters to an array list of tokens, which stores the token type, string and a pointer to the next token on an array. Once the scan reaches an EOF character, it will return the array list of tokens.

Parameters

∗text	is a string or file that is read by the scanner

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

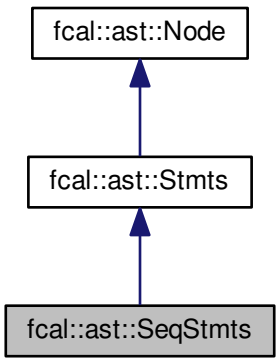
- include/scanner.h
- src/scanner.cc

4.43 fcal::ast::SeqStmts Class Reference

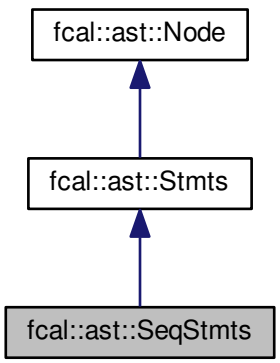
The SeqStmts class inherits directly from the abstract Stmts parent class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::SeqStmts:



Collaboration diagram for fcal::ast::SeqStmts:



Public Member Functions

- SeqStmts (Stmt *stmt, Stmts *stmts)
- std::string unparse ()

  SeqStmts unprase() method.
- std::string **cpp_code** ()

4.43.1 Detailed Description

The SeqStmts class inherits directly from the abstract Stmts parent class.

4.43.2 Constructor & Destructor Documentation

4.43.2.1 fcal::ast::SeqStmts::SeqStmts (Stmt * *stmt*, Stmts * *stmts*) [inline],[explicit]

SeqStmts production class takes two paramters: *stmt, and *stmts

Parameters

**stmt*	the statement on the LHS within the sequence of statements
**stmts*	the statements on the RHS within the sequence of statements

4.43.3 Member Function Documentation

4.43.3.1 std::string fcal::ast::SeqStmts::unparse ( ) [virtual]

SeqStmts unprase() method.

SeqStmts unparse() returns the stmt_ and stmts_ parameters.

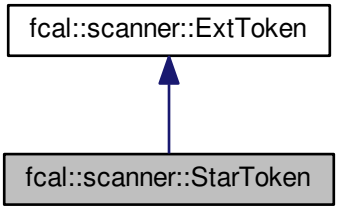
Implements fcal::ast::Node.

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

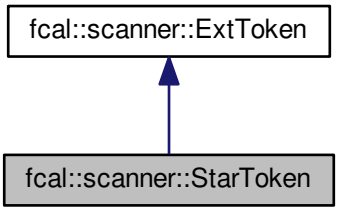
- include/ast.h
- src/ast.cc

4.44 fcal::scanner::StarToken Class Reference

Inheritance diagram for fcal::scanner::StarToken:



Collaboration diagram for fcal::scanner::StarToken:



Public Member Functions

- **StarToken** (parser::Parser ∗p, Token ∗t)
- parser::ParseResult **led** (parser::ParseResult left)
- std::string **description** ()
- int **lbp** ()

Additional Inherited Members

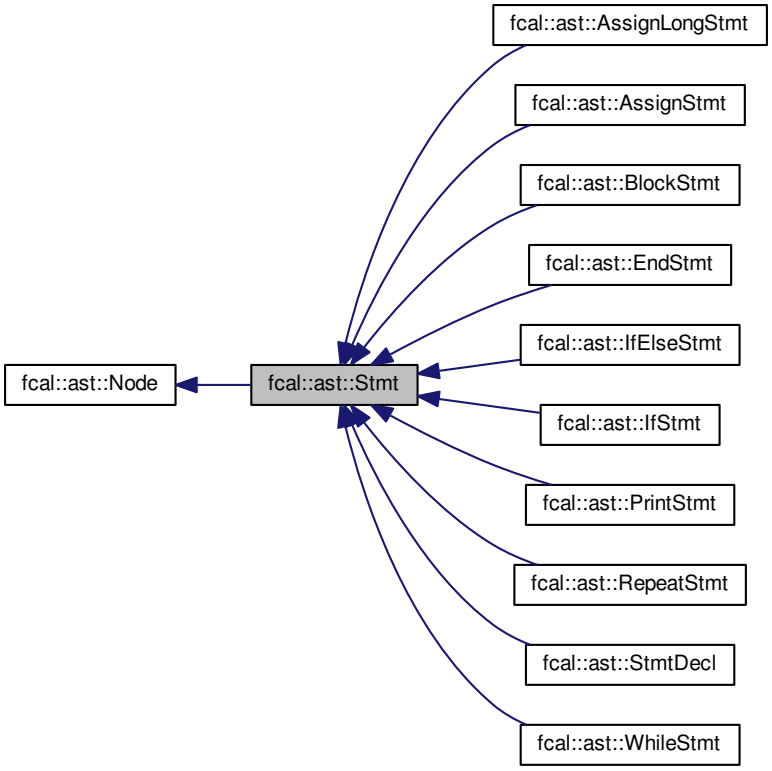
The documentation for this class was generated from the following file:

- include/ext_token.h

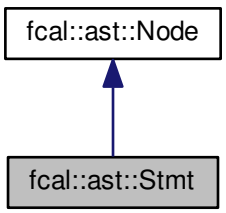
4.45 fcal::ast::Stmt Class Reference

`#include <ast.h>`

Inheritance diagram for fcal::ast::Stmt:



Collaboration diagram for fcal::ast::Stmt:



Additional Inherited Members

4.45.1 Detailed Description

This is an abstract Stmt class that inherits directly from the parent Node class.

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

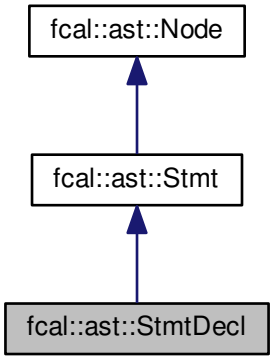
- include/ast.h

4.46 fcal::ast::StmtDecl Class Reference

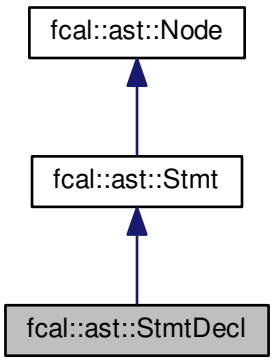
The StmtDecl class inherits directly from the abstract Stmt parent class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::StmtDecl:



Collaboration diagram for fcal::ast::StmtDecl:



Public Member Functions

- StmtDecl (Decl *decl)
- std::string unparse ()

  StmtDecl unparse() method.
- std::string **cpp_code** ()

4.46.1 Detailed Description

The StmtDecl class inherits directly from the abstract Stmt parent class.

4.46.2 Constructor & Destructor Documentation

4.46.2.1 fcal::ast::StmtDecl::StmtDecl (Decl * *decl*) [inline],[explicit]

StmtDecl production class takes a single parameter: decl

Parameters

**decl*	is a declaration found with in a single statement

4.46.3 Member Function Documentation

4.46.3.1 std::string fcal::ast::StmtDecl::unparse ( ) [virtual]

StmtDecl unparse() method.

StmtDecl unparse() returns the decl_ parameter.

Implements fcal::ast::Node.

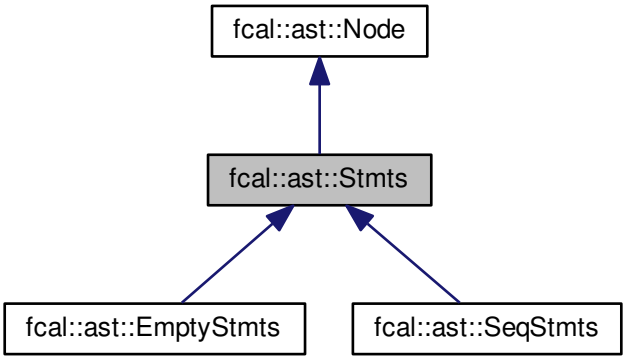
The documentation for this class was generated from the following files:

- include/ast.h
- src/ast.cc

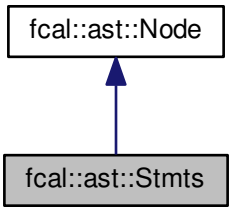
4.47 fcal::ast::Stmts Class Reference

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::Stmts:



Collaboration diagram for fcal::ast::Stmts:



Additional Inherited Members

4.47.1 Detailed Description

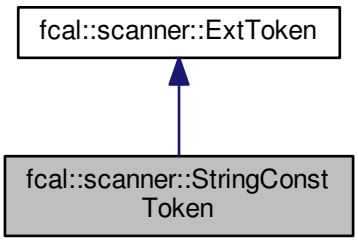
This is an abstract Stmts class that inherits directly from the parent Node class.

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

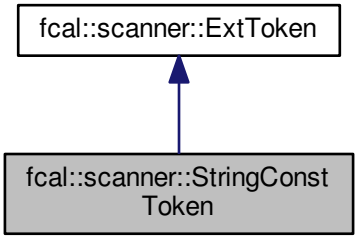
- include/ast.h

4.48 fcal::scanner::StringConstToken Class Reference

Inheritance diagram for fcal::scanner::StringConstToken:



Collaboration diagram for fcal::scanner::StringConstToken:



Public Member Functions

- **StringConstToken** (parser::Parser ∗p, Token ∗t)
- parser::ParseResult **nud** ()
- std::string **description** ()

Additional Inherited Members

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

- include/ext_token.h

4.49 fcal::scanner::Token Class Reference

`#include <scanner.h>`

Public Member Functions

- Token ()

  Token() constructor.
- Token (TokenType terminal, std::string lexeme, Token *next)

  Token parameters: terminal, lexeme and next.
- ~Token ()

  Token() deconstructor.
- TokenType get_terminal_ ()

  Token terminal_ accessor method.
- void set_terminal_ (TokenType terminal)

  Token terminal_ mutator method.
- std::string get_lexeme_ ()

  Token lexeme_ accessor method.
- void set_lexeme_ (std::string lexeme)

  Token lexeme_ mutator method.
- Token * get_next_ ()

  Token next_ accessor method.
- void set_next_ (Token *next)

  Token next_ mutator method.
- void **return_terminal_** ()
- void **return_lexeme_** ()
- void **return_next_** ()

Public Attributes

- int **length_of_lexeme_**

4.49.1 Detailed Description

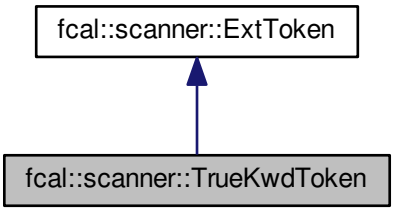
The Token class defines the various methods and attributes associated with the token types listed in the enum kTokenEnumType.

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

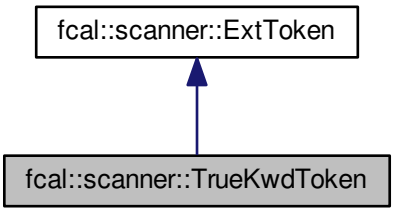
- include/scanner.h
- src/scanner.cc

4.50 fcal::scanner::TrueKwdToken Class Reference

Inheritance diagram for fcal::scanner::TrueKwdToken:



Collaboration diagram for fcal::scanner::TrueKwdToken:



Public Member Functions

- **TrueKwdToken** (parser::Parser ∗p, Token ∗t)
- parser::ParseResult **nud** ()
- std::string **description** ()

Additional Inherited Members

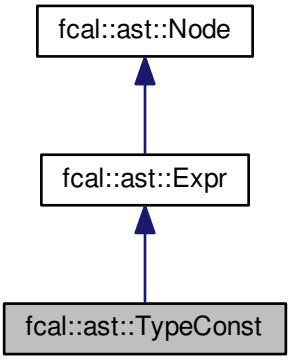
The documentation for this class was generated from the following file:

- include/ext_token.h

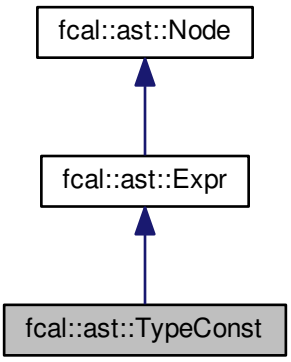
4.51 fcal::ast::TypeConst Class Reference

`#include <ast.h>`

Inheritance diagram for fcal::ast::TypeConst:



Collaboration diagram for fcal::ast::TypeConst:



Public Member Functions

- TypeConst (std::string type_const)
- std::string unparse ()

  TypeConst unparse() method.
- std::string **cpp_code** ()

4.51.1 Detailed Description

The TypeConst class inherits directly from the parent Expr class. The TypeConst class combines the redundant nature of the implementing multiple production rule classes to define integer, float, and string constants.

The integer, float, and string constants are defined by referencing the lexeme member of the prev_token_.

4.51.2 Constructor & Destructor Documentation

4.51.2.1 fcal::ast::TypeConst::TypeConst (std::string *type_const*) [inline], [explicit]

TypeConst production rules take the parameter: type_const

Parameters

type_const	refers to the constant of a given data type

4.51.3 Member Function Documentation

4.51.3.1 std::string fcal::ast::TypeConst::unparse ( ) [virtual]

TypeConst unparse() method.

TypeConst returns the type_const of a data type.

Implements fcal::ast::Node.

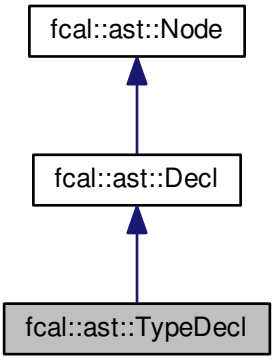
The documentation for this class was generated from the following files:

- include/ast.h
- src/ast.cc

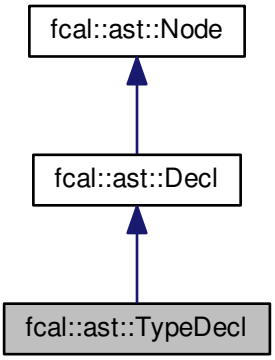
4.52 fcal::ast::TypeDecl Class Reference

```
#include <ast.h>
```

Inheritance diagram for fcal::ast::TypeDecl:



Collaboration diagram for fcal::ast::TypeDecl:



Public Member Functions

- TypeDecl (VarName *type, VarName *var_name)
- std::string unparse ()

  TypeDecl unparse() method.
- std::string **cpp_code** ()

4.52.1 Detailed Description

The TypeDecl class inherits directly from the parent Decl class. The TypeDecl class combines the redundant nature of the implementing multiple production rule classes to define integer, float, string and boolean data types.

The integer, float, string and data types is defined by the *type parameter that is passed to the constructor.

4.52.2 Constructor & Destructor Documentation

4.52.2.1 fcal::ast::TypeDecl::TypeDecl (VarName * *type*, VarName * *var_name*) [inline], [explicit]

TypeDecl production class takes the parameters: *type and *var_name

Parameters

**type*	defines the data type of the declaration
**var_name*	defines the variable name of the specific data type

4.52.3 Member Function Documentation

4.52.3.1 std::string fcal::ast::TypeDecl::unparse ( ) [virtual]

TypeDecl unparse() method.

TypeDecl returns the type_ and var_name_ of a declaration.

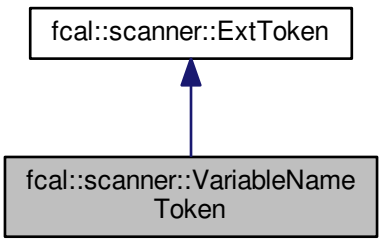
Implements fcal::ast::Node.

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

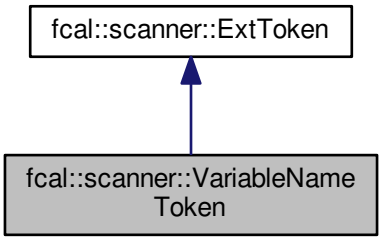
- include/ast.h
- src/ast.cc

4.53 fcal::scanner::VariableNameToken Class Reference

Inheritance diagram for fcal::scanner::VariableNameToken:



Collaboration diagram for fcal::scanner::VariableNameToken:



Public Member Functions

- **VariableNameToken** (parser::Parser ∗p, Token ∗t)
- parser::ParseResult **nud** ()
- std::string **description** ()

Additional Inherited Members

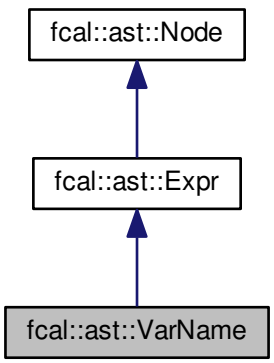
The documentation for this class was generated from the following file:

- include/ext_token.h

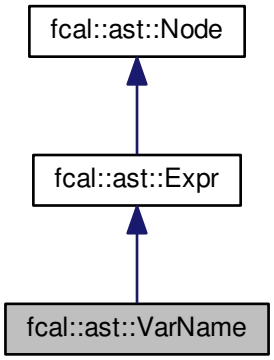
4.54 fcal::ast::VarName Class Reference

`#include <ast.h>`

Inheritance diagram for fcal::ast::VarName:



Collaboration diagram for fcal::ast::VarName:



Public Member Functions

- VarName (std::string lexeme)
- std::string unparse ()

  VarName unparse() method.
- std::string **cpp_code** ()

4.54.1 Detailed Description

The VarName class is actually a terminal type that is constantly referenced by the other production rules. The VarName class is defined as a child of the Expr only because of the specifications of the production rules. Otherwise, VarName could also be defined as a class inheriting directly from the Node class.

4.54.2 Constructor & Destructor Documentation

4.54.2.1 fcal::ast::VarName::VarName (std::string *lexeme*) [inline], [explicit]

VarName production rules take the parameter: lexeme

Parameters

lexeme	is the lexeme string of a given token

4.54.3 Member Function Documentation

4.54.3.1 std::string fcal::ast::VarName::unparse ( ) [virtual]

VarName unparse() method.

VarName unparse() returns the lexeme_ parameter.

Implements fcal::ast::Node.

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

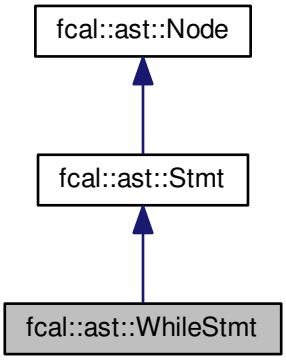
- include/ast.h
- src/ast.cc

4.55 fcal::ast::WhileStmt Class Reference

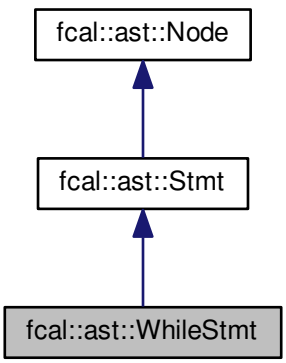
The WhileStmt class inherits directly from the abstract Stmt parent class.

`#include <ast.h>`

Inheritance diagram for fcal::ast::WhileStmt:



Collaboration diagram for fcal::ast::WhileStmt:



Public Member Functions

- WhileStmt (Expr *expr, Stmt *stmt)
- std::string unparse ()

  WhileStmt unparse() method.
- std::string **cpp_code** ()

4.55.1 Detailed Description

The WhileStmt class inherits directly from the abstract Stmt parent class.

4.55.2 Constructor & Destructor Documentation

4.55.2.1 fcal::ast::WhileStmt::WhileStmt (Expr * *expr*, Stmt * *stmt*) [inline], [explicit]

WhileStmt production class takes the parameters: *expr and *stmt

Parameters

**expr*	the expression being evaluated to continue looping
**stmt*	the statement to be looped

4.55.3 Member Function Documentation

4.55.3.1 std::string fcal::ast::WhileStmt::unparse ( ) [virtual]

WhileStmt unparse() method.

WhileStmt returns expr_ and stmt_ parameters.

Implements fcal::ast::Node.

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

- include/ast.h
- src/ast.cc

Index