

My Project

Generated by Doxygen 1.8.17

1 ParaCL	1
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 Namespace Documentation	9
5.1 yy Namespace Reference	9
5.1.1 Detailed Description	9
6 Class Documentation	11
6.1 AST::ANDNode Class Reference	11
6.1.1 Member Function Documentation	12
6.1.1.1 calc()	12
6.2 AST::ASNode Class Reference	13
6.2.1 Detailed Description	14
6.3 AST::CNode Class Reference	14
6.3.1 Detailed Description	15
6.3.2 Constructor & Destructor Documentation	15
6.3.2.1 CNode()	15
6.3.3 Member Function Documentation	15
6.3.3.1 calc()	15
6.4 yy::Driver Class Reference	15
6.4.1 Member Function Documentation	16
6.4.1.1 yylex()	16
6.5 AST::DVNode Class Reference	16
6.5.1 Member Function Documentation	17
6.5.1.1 calc()	17
6.6 AST::EQNode Class Reference	18
6.6.1 Member Function Documentation	19
6.6.1.1 calc()	19
6.7 AST::GENode Class Reference	19
6.7.1 Member Function Documentation	20
6.7.1.1 calc()	20
6.8 AST::GNode Class Reference	21
6.8.1 Member Function Documentation	22
6.8.1.1 calc()	22
6.9 AST::IFNode Class Reference	22
6.9.1 Detailed Description	23

6.9.2 Constructor & Destructor Documentation	23
6.9.2.1 ~IFNode()	23
6.9.3 Member Function Documentation	23
6.9.3.1 calc()	24
6.10 AST::INode Struct Reference	25
6.11 AST::IScope Struct Reference	26
6.12 AST::LENode Class Reference	27
6.12.1 Member Function Documentation	28
6.12.1.1 calc()	28
6.13 AST::LNode Class Reference	28
6.13.1 Member Function Documentation	29
6.13.1.1 calc()	29
6.14 AST::MLNode Class Reference	30
6.14.1 Member Function Documentation	31
6.14.1.1 calc()	31
6.15 AST::NEQNode Class Reference	31
6.15.1 Member Function Documentation	32
6.15.1.1 calc()	32
6.16 AST::OPNode Class Reference	33
6.16.1 Detailed Description	34
6.16.2 Constructor & Destructor Documentation	34
6.16.2.1 OPNode()	34
6.16.2.2 ~OPNode()	35
6.17 AST::ORNode Class Reference	35
6.17.1 Member Function Documentation	36
6.17.1.1 calc()	36
6.18 OurFlexLexer Class Reference	37
6.19 AST::PLNode Class Reference	38
6.19.1 Member Function Documentation	39
6.19.1.1 calc()	39
6.20 AST::PNode Class Reference	39
6.20.1 Detailed Description	40
6.20.2 Member Function Documentation	40
6.20.2.1 calc()	40
6.21 AST::RNode Class Reference	40
6.21.1 Detailed Description	41
6.21.2 Member Function Documentation	41
6.21.2.1 calc()	41
6.22 AST::SBNode Class Reference	42
6.22.1 Member Function Documentation	43
6.22.1.1 calc()	43
6.23 AST::Scope Class Reference	43

6.23.1 Constructor & Destructor Documentation	44
6.23.1.1 ~Scope()	44
6.23.2 Member Function Documentation	44
6.23.2.1 calc()	45
6.23.2.2 check_n_insert()	45
6.23.2.3 check_var()	45
6.23.2.4 loc_check()	46
6.23.2.5 push()	46
6.24 AST::VNode Class Reference	46
6.24.1 Detailed Description	47
6.24.2 Constructor & Destructor Documentation	47
6.24.2.1 VNode()	48
6.24.3 Member Function Documentation	48
6.24.3.1 calc()	48
6.24.3.2 get_loc()	48
6.24.3.3 get_name()	48
6.24.3.4 set_val()	48
6.25 AST::WHNode Class Reference	49
6.25.1 Detailed Description	50
6.25.2 Constructor & Destructor Documentation	50
6.25.2.1 ~WHNode()	50
6.25.3 Member Function Documentation	50
6.25.3.1 calc()	50
Index	51

Chapter 1

ParaCL

Programming C-like language for iLab 2nd course. **YOU SHOULD USE MAKEFILE**

by **Derzhavin Andrey** && **Khaidari Farid** && **Shurygin Anton**.

Chapter 2

Namespace Index

2.1 Namespace List

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

yy	<code>#include "numgrammar.tab.hh" - auto generated file from bison</code>	9
--------------------	--	-------------------

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

yy::Driver	15
AST::INode	25
AST::ASNode	13
AST::CNode	14
AST::IFNode	22
AST::IScope	26
AST::Scope	43
AST::OPNode	33
AST::ANDNode	11
AST::DVNode	16
AST::EQNode	18
AST::GENode	19
AST::GNode	21
AST::LENode	27
AST::LNode	28
AST::MLNode	30
AST::NEQNode	31
AST::ORNode	35
AST::PLNode	38
AST::SBNode	42
AST::PNode	39
AST::RNode	40
AST::VNode	46
AST::WHNode	49
yyFlexLexer	
OurFlexLexer	37

Chapter 4

Class Index

4.1 Class List

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

AST::ANDNode	11
AST::ASNode	
Assignment operator's node	13
AST::CNode	
Constant node class	14
yy::Driver	15
AST::DVNode	16
AST::EQNode	18
AST::GENode	19
AST::GNode	21
AST::IFNode	
If node class	22
AST::INode	25
AST::IScope	26
AST::LENode	27
AST::LNode	28
AST::MLNode	30
AST::NEQNode	31
AST::OPNode	
Operator node class	33
AST::ORNode	35
OurFlexLexer	37
AST::PLNode	38
AST::PNode	39
AST::RNode	40
AST::SBNode	42
AST::Scope	43
AST::VNode	
Variable node class	46
AST::WHNode	
While node class	49

Chapter 5

Namespace Documentation

5.1 yy Namespace Reference

#include "numgrammar.tab.hh" - auto generated file from bison

Classes

- class [Driver](#)

5.1.1 Detailed Description

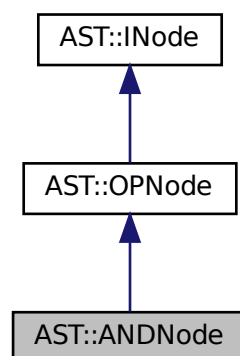
#include "numgrammar.tab.hh" - auto generated file from bison

Chapter 6

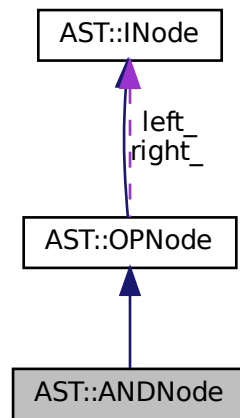
Class Documentation

6.1 AST::ANDNode Class Reference

Inheritance diagram for AST::ANDNode:



Collaboration diagram for AST::ANDNode:



Public Member Functions

- **ANDNode** (INode *left, INode *right)
- **ANDNode** (const ANDNode &)=delete
- **ANDNode** & **operator=** (const ANDNode &)=delete
- int **calc** () const override
Logical AND.

Additional Inherited Members

6.1.1 Member Function Documentation

6.1.1.1 calc()

```
int AST::ANDNode::calc ( ) const [override], [virtual]
```

Logical AND.

Returns

1 or 0

Implements [AST::INode](#).

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

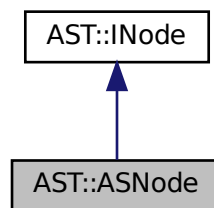
- AST/OPNode.hh
- AST/OPNode.cc

6.2 AST::ASNode Class Reference

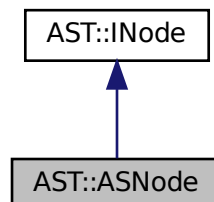
Assignment operator's node.

```
#include <OPNode.hh>
```

Inheritance diagram for AST::ASNode:



Collaboration diagram for AST::ASNode:



Public Member Functions

- **ASNode** (VNode *dst, INode *expr)
- **ASNode** (const ASNode &)=delete
- **ASNode** & **operator=** (const ASNode &)=delete
- int **calc** () const override
Interpret the node function.
- **~ASNode** ()
Assignment class destructor.

6.2.1 Detailed Description

Assignment operator's node.

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

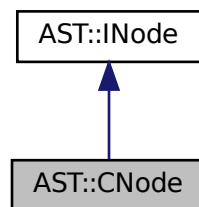
- AST/OPNode.hh
- AST/OPNode.cc

6.3 AST::CNode Class Reference

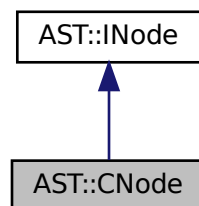
Constant node class.

```
#include <Node.hh>
```

Inheritance diagram for AST::CNode:



Collaboration diagram for AST::CNode:



Public Member Functions

- [CNode](#) (int val)
- [CNode](#) (const [CNode](#) &)=delete
- [CNode](#) & [operator=](#) (const [CNode](#) &)=delete
- int [calc](#) () const override

6.3.1 Detailed Description

Constant node class.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 CNode()

```
AST::CNode::CNode (
    int val )
```

Constant node ctor

Parameters

<i>val</i>	[in] - value of a node
------------	------------------------

6.3.3 Member Function Documentation

6.3.3.1 calc()

```
int AST::CNode::calc ( ) const [override], [virtual]
```

Calculate the value of node

Returns

value of a node

Implements [AST::INode](#).

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

- AST/Node.hh
- AST/Node.cc

6.4 yy::Driver Class Reference

Public Member Functions

- **Driver** (const char *name_of_file)
- **Driver** (const [Driver](#) &drv)=delete
- [Driver](#) & **operator=** (const [Driver](#) &)=delete
- bool **parse** ()
- parser::token_type [yylex](#) (parser::semantic_type *yyval, parser::location_type *yylloc)

6.4.1 Member Function Documentation

6.4.1.1 yylex()

```
yy::parser::token_type yy::Driver::yylex (
    parser::semantic_type * yyval,
    parser::location_type * yyloc )
```

The lexical analyzer function, `yylex`, recognizes tokens from the input stream and returns them to the parser.

Parameters

<i>yyval</i>	
<i>yyloc</i>	

Returns

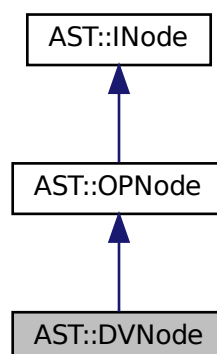
token type

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

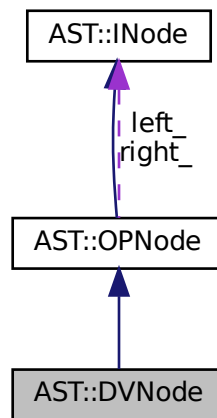
- `driver/driver.hh`
- `driver/driver.cc`

6.5 AST::DVNode Class Reference

Inheritance diagram for `AST::DVNode`:



Collaboration diagram for AST::DVNode:



Public Member Functions

- **DVNode** ([INode](#) *left, [INode](#) *right)
- **DVNode** (const [DVNode](#) &)=delete
- **DVNode** & **operator=** (const [DVNode](#) &)=delete
- int [calc](#) () const override

Additional Inherited Members

6.5.1 Member Function Documentation

6.5.1.1 calc()

```
int AST::DVNode::calc ( ) const [override], [virtual]
```

Calculate value of node function

Returns

calculation result

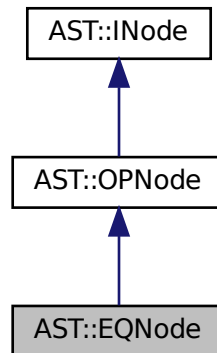
Implements [AST::INode](#).

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

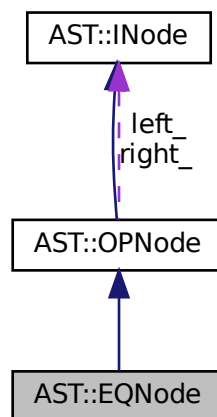
- AST/OPNode.hh
- AST/OPNode.cc

6.6 AST::EQNode Class Reference

Inheritance diagram for AST::EQNode:



Collaboration diagram for AST::EQNode:



Public Member Functions

- **EQNode** ([INode](#) *left, [INode](#) *right)
 - **EQNode** (const [EQNode](#) &)=delete
 - [EQNode](#) & **operator=** (const [EQNode](#) &)=delete
 - int [calc](#) () const override
- Calculate for equality.*

Additional Inherited Members

6.6.1 Member Function Documentation

6.6.1.1 calc()

```
int AST::EQNode::calc ( ) const [override], [virtual]
```

Calculate for equality.

Returns

1 if lhs is equal to rhs else 0

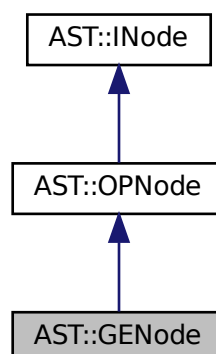
Implements [AST::INode](#).

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

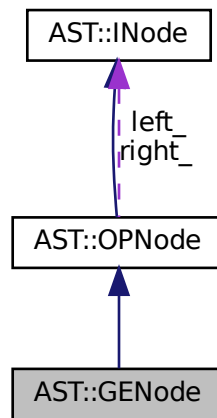
- AST/OPNode.hh
- AST/OPNode.cc

6.7 AST::GENode Class Reference

Inheritance diagram for AST::GENode:



Collaboration diagram for AST::GENode:



Public Member Functions

- **GENode** ([INode](#) *left, [INode](#) *right)
- **GENode** (const [GENode](#) &)=delete
- [GENode](#) & **operator=** (const [GENode](#) &)=delete
- int [calc](#) () const override
Checking for greatness or equality.

Additional Inherited Members

6.7.1 Member Function Documentation

6.7.1.1 calc()

```
int AST::GENode::calc ( ) const [override], [virtual]
```

Checking for greatness or equality.

Returns

1 if lhs is greater or equal then rhs

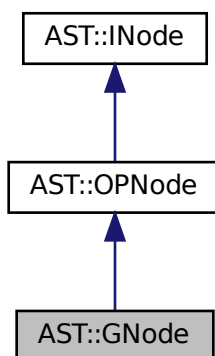
Implements [AST::INode](#).

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

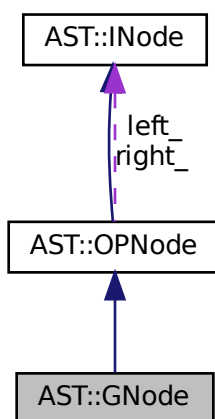
- AST/OPNode.hh
- AST/OPNode.cc

6.8 AST::GNode Class Reference

Inheritance diagram for AST::GNode:



Collaboration diagram for AST::GNode:



Public Member Functions

- **GNode** (*INode* *left, *INode* *right)
 - **GNode** (const *GNode* &)=delete
 - *GNode* & **operator=** (const *GNode* &)=delete
 - int **calc** () const override
- Checking for greatness.*

Additional Inherited Members

6.8.1 Member Function Documentation

6.8.1.1 `calc()`

```
int AST::GNode::calc ( ) const [override], [virtual]
```

Checking for greatness.

Returns

1 if lhs is greater then rhs

Implements [AST::INode](#).

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

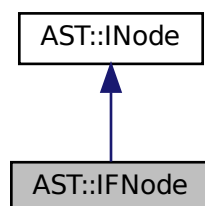
- AST/OPNode.hh
- AST/OPNode.cc

6.9 AST::IFNode Class Reference

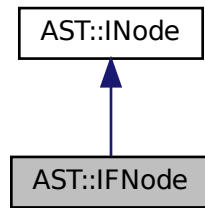
If node class.

```
#include <Node.hh>
```

Inheritance diagram for AST::IFNode:



Collaboration diagram for AST::IFNode:



Public Member Functions

- **IFNode** (*INode* *cond, *IScope* *if_sc, *IScope* *el_sc=nullptr)
- **IFNode** (const *IFNode* &)=delete
- *IFNode* & **operator=** (const *IFNode* &)=delete
- int *calc* () const override
- ~*IFNode* ()

6.9.1 Detailed Description

If node class.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 ~IFNode()

```
AST::IFNode::~~IFNode ( )
```

If node dtor function

6.9.3 Member Function Documentation

6.9.3.1 calc()

```
int AST::IFNode::calc ( ) const [override], [virtual]
```

Interpret If node function

Returns

calculated result

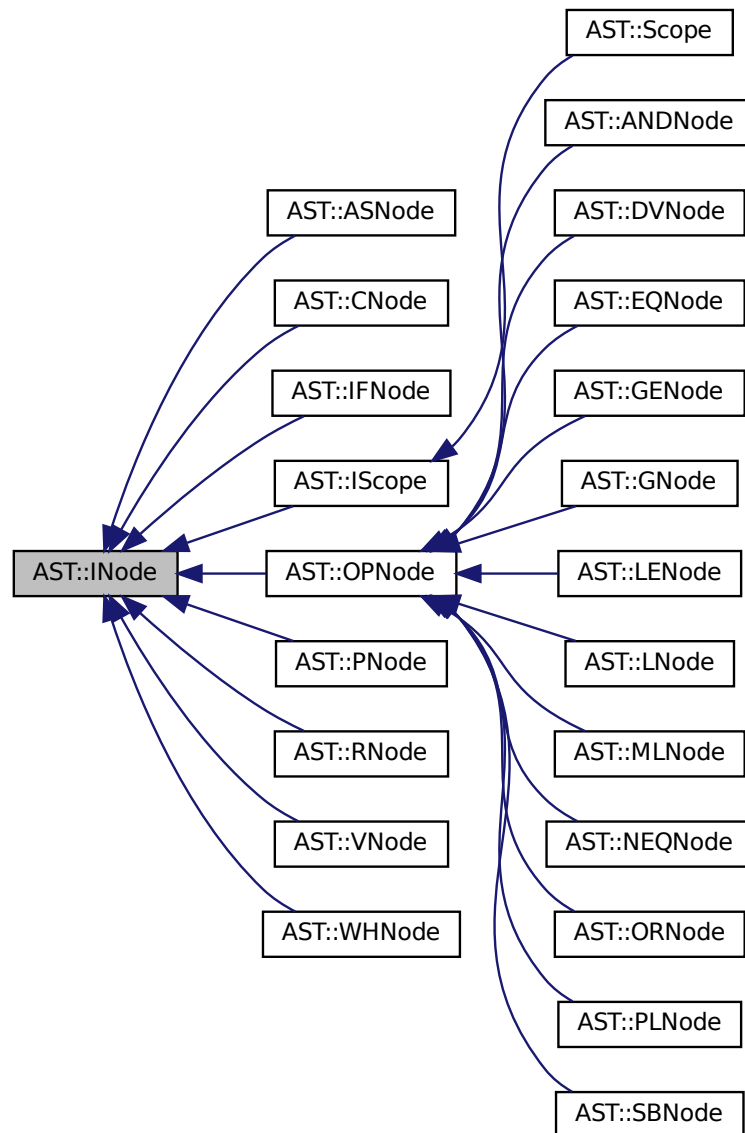
Implements [AST::INode](#).

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

- [AST/Node.hh](#)
- [AST/Node.cc](#)

6.10 AST::INode Struct Reference

Inheritance diagram for AST::INode:



Public Member Functions

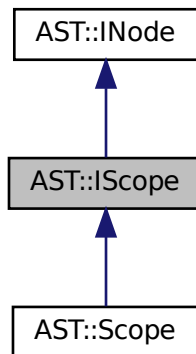
- virtual int **calc** () const =0

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

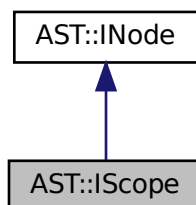
- AST/INode.hh

6.11 AST::IScope Struct Reference

Inheritance diagram for AST::IScope:



Collaboration diagram for AST::IScope:



Public Member Functions

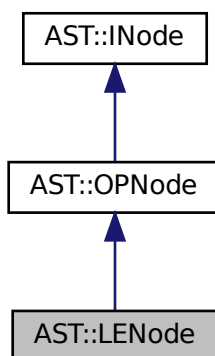
- virtual void **push** ([INode](#) *node)=0
- virtual [IScope](#) * **reset_scope** () const =0
- virtual std::pair< var_table::iterator, bool > **check_var** (const std::string &var_name)=0
- virtual std::pair< var_table::iterator, bool > **loc_check** (const std::string &var_name)=0
- virtual var_table::iterator **check_n_insert** (const std::string &var_name)=0

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

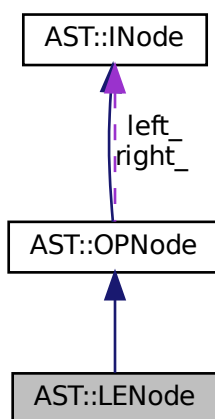
- AST/INode.hh

6.12 AST::LENode Class Reference

Inheritance diagram for AST::LENode:



Collaboration diagram for AST::LENode:



Public Member Functions

- **LENode** (*INode* *left, *INode* *right)
 - **LENode** (const *LENode* &)=delete
 - *LENode* & **operator=** (const *LENode* &)=delete
 - int *calc* () const override
- Checking for lessness or equality.*

Additional Inherited Members

6.12.1 Member Function Documentation

6.12.1.1 calc()

```
int AST::LNode::calc ( ) const [override], [virtual]
```

Checking for lessness or equality.

Returns

1 if lhs is less or equal then rhs

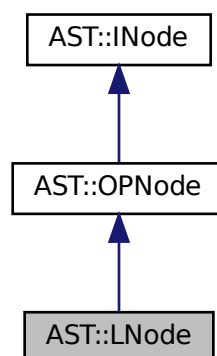
Implements [AST::INode](#).

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

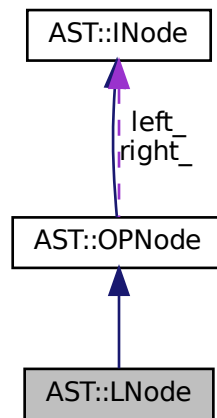
- AST/OPNode.hh
- AST/OPNode.cc

6.13 AST::LNode Class Reference

Inheritance diagram for AST::LNode:



Collaboration diagram for AST::LNode:



Public Member Functions

- **LNode** ([INode](#) *left, [INode](#) *right)
- **LNode** (const [LNode](#) &)=delete
- [LNode](#) & **operator=** (const [LNode](#) &)=delete
- int [calc](#) () const override
Checking for lessness.

Additional Inherited Members

6.13.1 Member Function Documentation

6.13.1.1 calc()

```
int AST::LNode::calc ( ) const [override], [virtual]
```

Checking for lessness.

Returns

1 if lhs is less then rhs

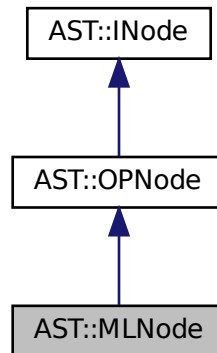
Implements [AST::INode](#).

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

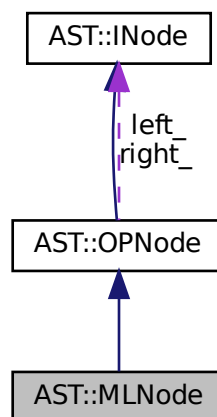
- AST/OPNode.hh
- AST/OPNode.cc

6.14 AST::MLNode Class Reference

Inheritance diagram for AST::MLNode:



Collaboration diagram for AST::MLNode:



Public Member Functions

- **MLNode** (*INode* *left, *INode* *right)
- **MLNode** (const *MLNode* &)=delete
- *MLNode* & **operator=** (const *MLNode* &)=delete
- int *calc* () const override

Additional Inherited Members

6.14.1 Member Function Documentation

6.14.1.1 calc()

```
int AST::MLNode::calc ( ) const [override], [virtual]
```

Calculate value of node function

Returns

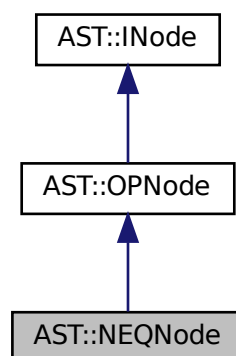
Implements [AST::INode](#).

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

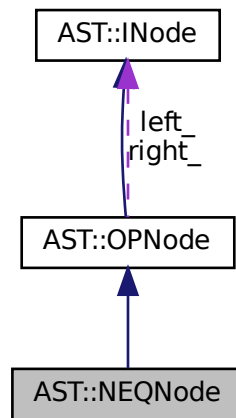
- AST/OPNode.hh
- AST/OPNode.cc

6.15 AST::NEQNode Class Reference

Inheritance diagram for AST::NEQNode:



Collaboration diagram for AST::NEQNode:



Public Member Functions

- **NEQNode** (INode *left, INode *right)
- **NEQNode** (const NEQNode &)=delete
- **NEQNode & operator=** (const NEQNode &)=delete
- int **calc** () const override

Calculate for not equality.

Additional Inherited Members

6.15.1 Member Function Documentation

6.15.1.1 calc()

```
int AST::NEQNode::calc ( ) const [override], [virtual]
```

Calculate for not equality.

Returns

1 if lhs is not equal to rhs else 0

Implements [AST::INode](#).

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

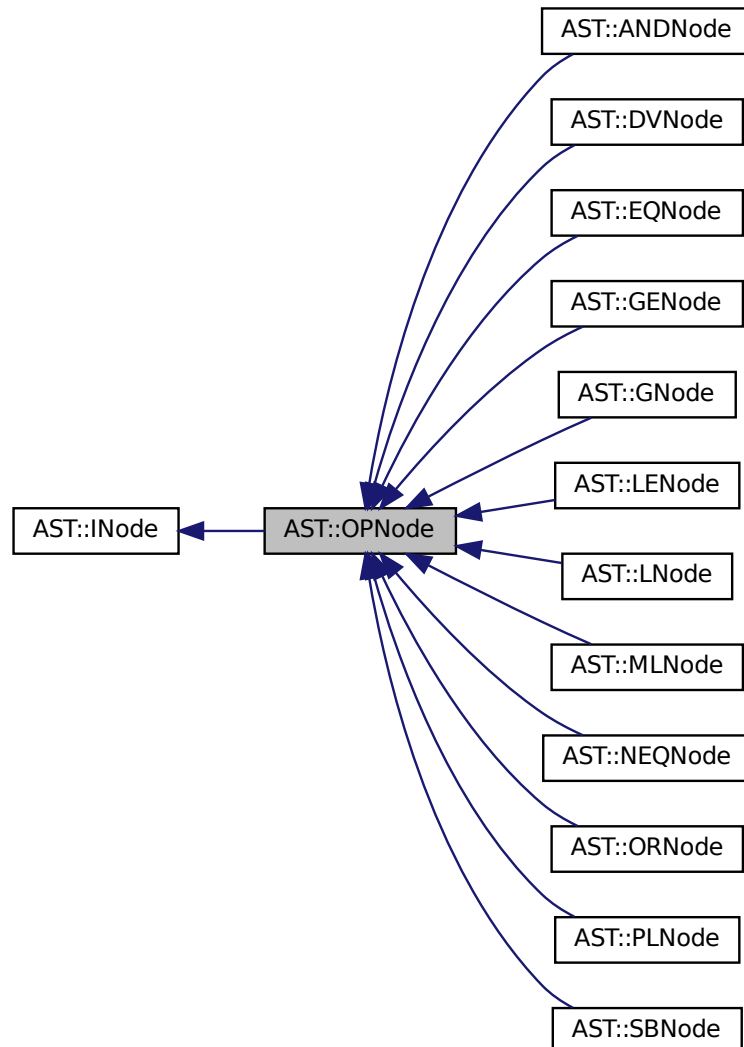
- AST/OPNode.hh
- AST/OPNode.cc

6.16 AST::OPNode Class Reference

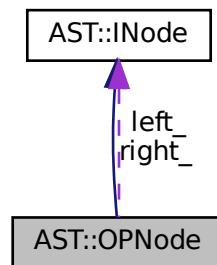
Operator node class.

```
#include <Node.hh>
```

Inheritance diagram for AST::OPNode:



Collaboration diagram for AST::OPNode:



Public Member Functions

- `OPNode (INode *left, INode *right)`
- `~OPNode ()`

Protected Attributes

- `INode * left_ {}`
- `INode * right_ {}`

6.16.1 Detailed Description

Operator node class.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 OPNode()

```

AST::OPNode::OPNode (
    INode * left,
    INode * right )
  
```

Operator's node constructor

Parameters

<i>left</i>	[in] - left node of operator
<i>right</i>	[in] - right node of operator

6.16.2.2 ~OPNode()

```
AST::OPNode::~~OPNode ( )
```

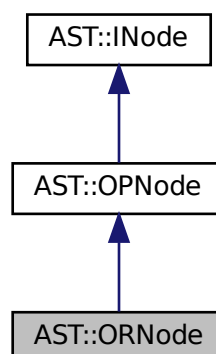
[OPNode](#) class destructor. Deletes left and right nodes

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

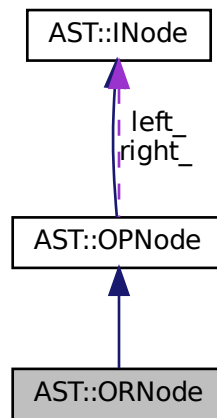
- [AST/Node.hh](#)
- [AST/Node.cc](#)

6.17 AST::ORNode Class Reference

Inheritance diagram for AST::ORNode:



Collaboration diagram for AST::ORNode:



Public Member Functions

- **ORNode** ([INode](#) *left, [INode](#) *right)
- **ORNode** (const [ORNode](#) &)=delete
- [ORNode](#) & **operator=** (const [ORNode](#) &)=delete
- int [calc](#) () const override
Logical OR.

Additional Inherited Members

6.17.1 Member Function Documentation

6.17.1.1 calc()

```
int AST::ORNode::calc ( ) const [override], [virtual]
```

Logical OR.

Returns

1 or 0

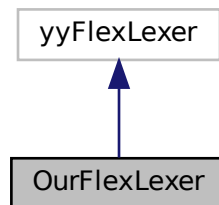
Implements [AST::INode](#).

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

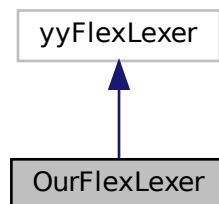
- AST/OPNode.hh
- AST/OPNode.cc

6.18 OurFlexLexer Class Reference

Inheritance diagram for OurFlexLexer:



Collaboration diagram for OurFlexLexer:



Public Member Functions

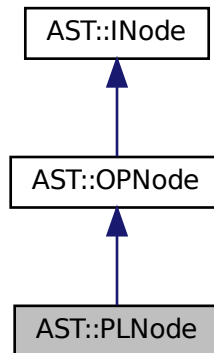
- **OurFlexLexer** (const [OurFlexLexer](#) &flx)=delete
- [OurFlexLexer](#) & **operator=** (const [OurFlexLexer](#) &)=delete
- yy::location **get_cur_location** ()
- std::string **get_cur_str** ()
- void **upd_cur_loc** ()
- int **yylex** () override

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

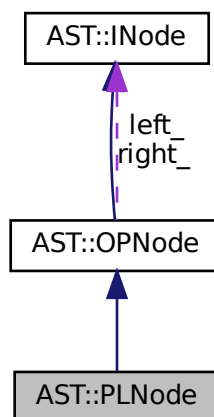
- parser/parser.hh
- parser/parser.cc

6.19 AST::PLNode Class Reference

Inheritance diagram for AST::PLNode:



Collaboration diagram for AST::PLNode:



Public Member Functions

- **PLNode** (*INode* *left, *INode* *right)
- **PLNode** (const *PLNode* &)=delete
- *PLNode* & **operator=** (const *PLNode* &)=delete
- int *calc* () const override

Additional Inherited Members

6.19.1 Member Function Documentation

6.19.1.1 calc()

```
int AST::PNode::calc ( ) const [override], [virtual]
```

Calculate value of node function

Returns

Implements [AST::INode](#).

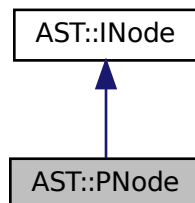
The documentation for this class was generated from the following files:

- AST/OPNode.hh
- AST/OPNode.cc

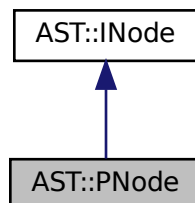
6.20 AST::PNode Class Reference

```
#include <Node.hh>
```

Inheritance diagram for AST::PNode:



Collaboration diagram for AST::PNode:



Public Member Functions

- **PNode** ([INode](#) *expr)
- **PNode** (const [PNode](#) &)=delete
- [PNode](#) & **operator=** (const [PNode](#) &)=delete
- int [calc](#) () const override

6.20.1 Detailed Description

Print node class function

6.20.2 Member Function Documentation

6.20.2.1 [calc\(\)](#)

```
int AST::PNode::calc ( ) const [override], [virtual]
```

Interpret print node function

Implements [AST::INode](#).

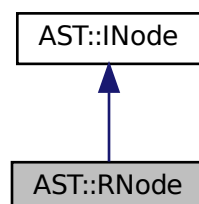
The documentation for this class was generated from the following files:

- [AST/Node.hh](#)
- [AST/Node.cc](#)

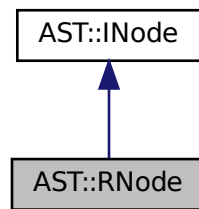
6.21 [AST::RNode](#) Class Reference

```
#include <Node.hh>
```

Inheritance diagram for [AST::RNode](#):



Collaboration diagram for AST::RNode:



Public Member Functions

- **RNode** (const [RNode](#) &)=delete
- [RNode](#) & **operator=** (const [RNode](#) &)=delete
- int [calc](#) () const override
Interpret read node function.

6.21.1 Detailed Description

Scanf node class

6.21.2 Member Function Documentation

6.21.2.1 calc()

```
int AST::RNode::calc ( ) const [override], [virtual]
```

Interpret read node function.

Returns

read value

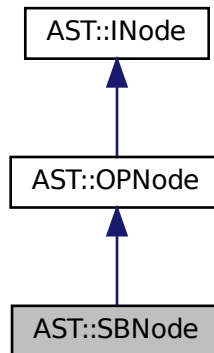
Implements [AST::INode](#).

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

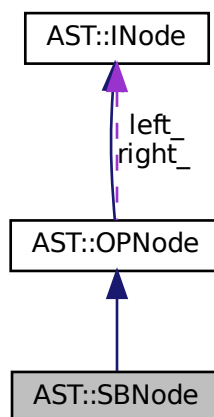
- AST/Node.hh
- AST/Node.cc

6.22 AST::SBNODE Class Reference

Inheritance diagram for AST::SBNODE:



Collaboration diagram for AST::SBNODE:



Public Member Functions

- **SBNODE** (*INode* *left, *INode* *right)
- **SBNODE** (const *SBNODE* &)=delete
- *SBNODE* & **operator=** (const *SBNODE* &)=delete
- int *calc* () const override

Additional Inherited Members

6.22.1 Member Function Documentation

6.22.1.1 calc()

```
int AST::SBNode::calc ( ) const [override], [virtual]
```

Calculate value of node function

Returns

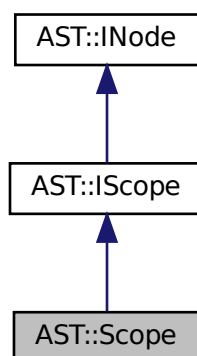
Implements [AST::INode](#).

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

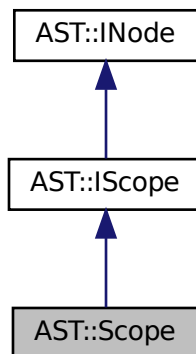
- AST/OPNode.hh
- AST/OPNode.cc

6.23 AST::Scope Class Reference

Inheritance diagram for AST::Scope:



Collaboration diagram for AST::Scope:



Public Member Functions

- **Scope** ([IScope](#) *parent=nullptr)
- **Scope** (const [Scope](#) &sc)=delete
- **Scope & operator=** (const [Scope](#) &sc)=delete
- **IScope * reset_scope** () const override
- int **calc** () const override
Interpret the scope function (claculate)
- void **push** ([INode](#) *node) override
Add node to scope function.
- it_bool **check_var** (const std::string &var_name) override
Check var in all available scopes function.
- it_bool **loc_check** (const std::string &var_name) override
Check variable in current scope function.
- var_table::iterator **check_n_insert** (const std::string &var_name) override
- **~Scope** () override

6.23.1 Constructor & Destructor Documentation

6.23.1.1 ~Scope()

```
AST::Scope::~Scope ( ) [override]
```

[Scope](#) class destructor

6.23.2 Member Function Documentation

6.23.2.1 calc()

```
int AST::Scope::calc ( ) const [override], [virtual]
```

Interpret the scope function (calculate)

Returns

int

Implements [AST::INode](#).

6.23.2.2 check_n_insert()

```
var_table::iterator AST::Scope::check_n_insert (
    const std::string & var_name ) [override], [virtual]
```

Parameters

<i>var_name</i>	
-----------------	--

Returns

Implements [AST::IScope](#).

6.23.2.3 check_var()

```
std::pair< var_table::iterator, bool > AST::Scope::check_var (
    const std::string & var_name ) [override], [virtual]
```

Check var in all available scopes function.

Parameters

<i>var_name</i>	[in] name of a var to get access to
-----------------	-------------------------------------

Returns

pair of iterator to var table and bool, which: TRUE - iterator is valid, variable found, FALSE - iterator is not valid (end()), variable was not found

Implements [AST::IScope](#).

6.23.2.4 loc_check()

```
Scope::it_bool AST::Scope::loc_check (
    const std::string & var_name ) [override], [virtual]
```

Check variable in current scope function.

Parameters

<i>var_name</i>	[in] name of a var to find
-----------------	----------------------------

Returns

pair of iterator to var table and bool, which: TRUE - iterator is valid, variable found, FALSE - iterator is not valid (end()), variable was not found

Implements [AST::IScope](#).

6.23.2.5 push()

```
void AST::Scope::push (
    INode * node ) [override], [virtual]
```

Add node to scope function.

Parameters

<i>node</i>	[in] node to add
-------------	------------------

Returns

none

Implements [AST::IScope](#).

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

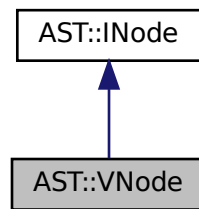
- AST/Node.hh
- AST/Node.cc

6.24 AST::VNode Class Reference

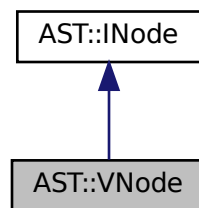
Variable node class.

```
#include <Node.hh>
```

Inheritance diagram for AST::VNode:



Collaboration diagram for AST::VNode:



Public Member Functions

- [VNode](#) (var_table::iterator loc)
- [VNode](#) (const [VNode](#) &)=delete
- [VNode](#) & **operator=** (const [VNode](#) &)=delete
- const std::string & [get_name](#) () const
- var_table::iterator [get_loc](#) () const
- int [calc](#) () const override
- void [set_val](#) (int val)

6.24.1 Detailed Description

Variable node class.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 VNode()

```
AST::VNode::VNode (
    var_table::iterator loc ) [explicit]
```

Varibale node ctor

Parameters

<i>name</i>	[in] name of a variable
<i>loc</i>	

6.24.3 Member Function Documentation

6.24.3.1 calc()

```
int AST::VNode::calc ( ) const [override], [virtual]
```

Calculate value of a variable function

Returns

Implements [AST::INode](#).

6.24.3.2 get_loc()

```
var_table::iterator AST::VNode::get_loc ( ) const
```

Get variable location in table function

Returns

6.24.3.3 get_name()

```
const std::string & AST::VNode::get_name ( ) const
```

Get variable name function

Returns

6.24.3.4 set_val()

```
void AST::VNode::set_val (
    int val )
```

Set value of variable function

Parameters

<i>val</i>	
------------	--

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

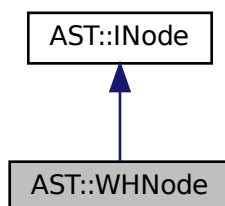
- AST/Node.hh
- AST/Node.cc

6.25 AST::WHNode Class Reference

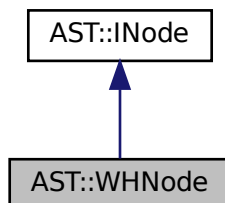
While node class.

```
#include <Node.hh>
```

Inheritance diagram for AST::WHNode:



Collaboration diagram for AST::WHNode:



Public Member Functions

- **WHNode** ([INode](#) *cond, [IScope](#) *scope)
- **WHNode** (const [WHNode](#) &)=delete
- [WHNode](#) & **operator=** (const [WHNode](#) &)=delete
- int [calc](#) () const override
Calculate while node function.
- [~WHNode](#) ()

6.25.1 Detailed Description

While node class.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 [~WHNode\(\)](#)

```
AST::WHNode::~~WHNode ( )
```

While node class destructor

6.25.3 Member Function Documentation

6.25.3.1 [calc\(\)](#)

```
int AST::WHNode::calc ( ) const [override], [virtual]
```

Calculate while node function.

Returns

int

Implements [AST::INode](#).

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

- [AST/Node.hh](#)
- [AST/Node.cc](#)

Index

- ~IFNode
 - AST::IFNode, 23
- ~OPNode
 - AST::OPNode, 35
- ~Scope
 - AST::Scope, 44
- ~WHNode
 - AST::WHNode, 50
- AST::ANDNode, 11
 - calc, 12
- AST::ASNode, 13
- AST::CNode, 14
 - calc, 15
 - CNode, 15
- AST::DVNode, 16
 - calc, 17
- AST::EQNode, 18
 - calc, 19
- AST::GENode, 19
 - calc, 20
- AST::GNode, 21
 - calc, 22
- AST::IFNode, 22
 - ~IFNode, 23
 - calc, 23
- AST::INode, 25
- AST::IScope, 26
- AST::LENode, 27
 - calc, 28
- AST::LNode, 28
 - calc, 29
- AST::MLNode, 30
 - calc, 31
- AST::NEQNode, 31
 - calc, 32
- AST::OPNode, 33
 - ~OPNode, 35
 - OPNode, 34
- AST::ORNode, 35
 - calc, 36
- AST::PLNode, 38
 - calc, 39
- AST::PNode, 39
 - calc, 40
- AST::RNode, 40
 - calc, 41
- AST::SBNode, 42
 - calc, 43
- AST::Scope, 43

- ~Scope, 44
- calc, 44
- check_n_insert, 45
- check_var, 45
- loc_check, 45
- push, 46
- AST::VNode, 46
 - calc, 48
 - get_loc, 48
 - get_name, 48
 - set_val, 48
 - VNode, 47
- AST::WHNode, 49
 - ~WHNode, 50
 - calc, 50
- calc
 - AST::ANDNode, 12
 - AST::CNode, 15
 - AST::DVNode, 17
 - AST::EQNode, 19
 - AST::GENode, 20
 - AST::GNode, 22
 - AST::IFNode, 23
 - AST::LENode, 28
 - AST::LNode, 29
 - AST::MLNode, 31
 - AST::NEQNode, 32
 - AST::ORNode, 36
 - AST::PLNode, 39
 - AST::PNode, 40
 - AST::RNode, 41
 - AST::SBNode, 43
 - AST::Scope, 44
 - AST::VNode, 48
 - AST::WHNode, 50
- check_n_insert
 - AST::Scope, 45
- check_var
 - AST::Scope, 45
- CNode
 - AST::CNode, 15
- get_loc
 - AST::VNode, 48
- get_name
 - AST::VNode, 48
- loc_check
 - AST::Scope, 45

OPNode
 AST::OPNode, [34](#)
OurFlexLexer, [37](#)

push
 AST::Scope, [46](#)

set_val
 AST::VNode, [48](#)

VNode
 AST::VNode, [47](#)

yy, [9](#)
yy::Driver, [15](#)
 yylex, [16](#)
yylex
 yy::Driver, [16](#)