

3 address

1.0

Generated by Doxygen 1.8.8

Mon Apr 13 2015 00:24:38



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	YYSTYPE::BackpatchList Struct Reference . . . . .	5
3.2	BackpatchList Struct Reference . . . . .	5
3.3	YYSTYPE::info Struct Reference . . . . .	6
3.4	info Struct Reference . . . . .	6
3.5	YYSTYPE::marks Struct Reference . . . . .	6
3.6	marks Struct Reference . . . . .	7
3.7	next Struct Reference . . . . .	7
3.8	YYSTYPE::next Struct Reference . . . . .	7
3.9	Quadruple Struct Reference . . . . .	8
3.10	YYSTYPE Union Reference . . . . .	8
<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	codegener.cpp File Reference . . . . .	9
4.1.1	Detailed Description . . . . .	10
4.1.2	Typedef Documentation . . . . .	10
4.1.2.1	patchList . . . . .	10
4.1.3	Function Documentation . . . . .	10
4.1.3.1	printCode . . . . .	10
4.1.3.2	gencode . . . . .	10
4.1.3.3	gencode . . . . .	11
4.1.3.4	gencode . . . . .	11
4.1.3.5	gencode . . . . .	11
4.1.3.6	gencode . . . . .	12
	<b>Index</b>	<b>13</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">YYSTYPE::BackpatchList</a>	5
<a href="#">BackpatchList</a>	5
<a href="#">YYSTYPE::info</a>	6
<a href="#">info</a>	6
<a href="#">YYSTYPE::marks</a>	6
<a href="#">marks</a>	7
<a href="#">next</a>	7
<a href="#">YYSTYPE::next</a>	7
<a href="#">Quadruple</a>	8
<a href="#">YYSTYPE</a>	8



## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">codegener.cpp</a>	This file includes grammar rule and their semantic action(s) . . . . .	9
-------------------------------	--	---





## Chapter 3

# Class Documentation

### 3.1 YYSYTYPE::BackpatchList Struct Reference

Collaboration diagram for YYSYTYPE::BackpatchList:

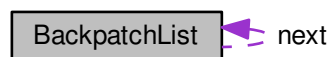


The documentation for this struct was generated from the following file:

- yaccrule.cpp

### 3.2 BackpatchList Struct Reference

Collaboration diagram for BackpatchList:

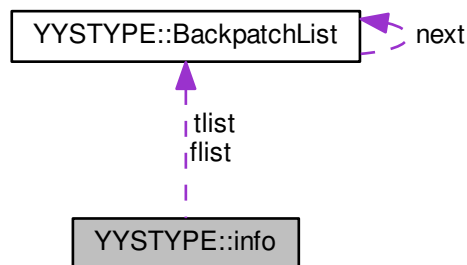


The documentation for this struct was generated from the following file:

- yaccrule.cpp

### 3.3 YYSTYPE::info Struct Reference

Collaboration diagram for YYSTYPE::info:

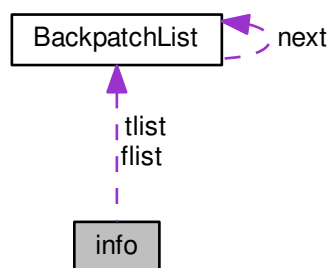


The documentation for this struct was generated from the following file:

- yaccrule.cpp

### 3.4 info Struct Reference

Collaboration diagram for info:



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

- yaccrule.cpp

### 3.5 YYSTYPE::marks Struct Reference

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

- yaccrule.cpp

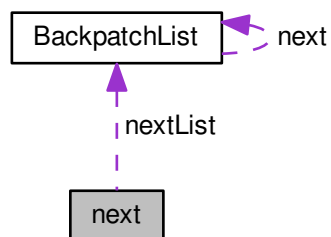
### 3.6 marks Struct Reference

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

- yaccrule.cpp

### 3.7 next Struct Reference

Collaboration diagram for next:

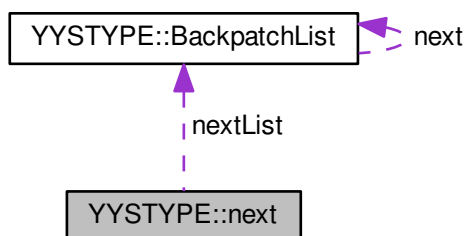


The documentation for this struct was generated from the following file:

- yaccrule.cpp

### 3.8 YYSTYPE::next Struct Reference

Collaboration diagram for YYSTYPE::next:



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

- yaccrule.cpp

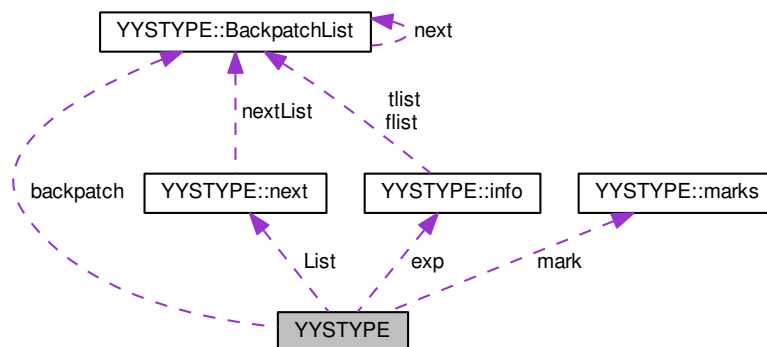
### 3.9 Quadruple Struct Reference

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

- [yacrule.cpp](#)
- [codegener.cpp](#)

### 3.10 YYSTYPE Union Reference

Collaboration diagram for YYSTYPE:



#### Classes

- struct [BackpatchList](#)
- struct [info](#)
- struct [marks](#)
- struct [next](#)

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

- [yacrule.cpp](#)

## Chapter 4

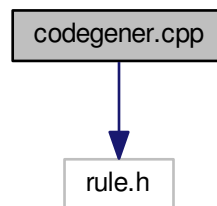
# File Documentation

### 4.1 codegener.cpp File Reference

This file includes grammar rule and their semantic action(s).

```
#include "rule.h"
```

Include dependency graph for codegener.cpp:



#### Classes

- struct [Quadruple](#)

#### Typedefs

- typedef [YYSTYPE::BackpatchList](#) [patchList](#)  
*generate new temporary variable name*

#### Functions

- void [printCode](#) ()  
*print code to the console after completion of parsing and store in file name 'output.txt' for future use*
- void [gencode](#) (const char \*result, const char \*addr1, const char \*op, const char \*addr2)  
*Generate 3 address code and store in Quadruple table for binary expression e.g.*
- void [gencode](#) (const char \*result, const char \*unop, const char \*addr1)

*Generate 3 address code and store in Quadrule table for unary expression e.g.*

- void `gencode` (const char \*result, const char \*addr1, const char \*op, const char \*addr2, const char \*addr3, int label)

*Generate 3 address code and store in Quadrule table for conditional jump.*

- void `gencode` (const char \*result, const char \*addr1)

*Generate 3 address code and store in Quadrule table for assignement e.g.*

- void `gencode` (const char \*result, int label)

*Generate 3 address code and store in Quadrule table for goto target e.g.*

#### 4.1.1 Detailed Description

This file includes grammar rule and their semantic action(s).

##### Version

1.0

#### 4.1.2 Typedef Documentation

##### 4.1.2.1 typedef YYSTYPE::BackpatchList patchList

generate new temporary variable name

##### Parameters

<i>void</i>	None
-------------	------

##### Returns

newtemp char\*

#### 4.1.3 Function Documentation

##### 4.1.3.1 void printCode ( )

print code to the console after completion of parsing and store in file name 'output.txt' for future use

##### Parameters

<i>void</i>	None
-------------	------

##### Returns

void None

##### 4.1.3.2 void gencode ( const char \* result, const char \* addr1, const char \* op, const char \* addr2 )

Generate 3 address code and store in Quadrule table for binary expression e.g.

a=t0+b;

## Parameters

<i>result</i>	char*
<i>address1</i>	char*
<i>binary_operator</i>	char*
<i>address2</i>	char*

## Returns

void None

4.1.3.3 void gencode ( const char \* *result*, const char \* *unop*, const char \* *addr1* )

Generate 3 address code and store in Quadrule table for unary expression e.g.

a = -b;

## Parameters

<i>result</i>	char*
<i>address1</i>	char*
<i>unary_operator</i>	char*

## Returns

void None

4.1.3.4 void gencode ( const char \* *result*, const char \* *addr1*, const char \* *op*, const char \* *addr2*, const char \* *addr3*, int *label* )

Generate 3 address code and store in Quadrule table for conditional jump.

## Parameters

<i>if</i>	char*
<i>address1</i>	char*
<i>relational_↔ operator</i>	char*
<i>address2</i>	char*
<i>goto</i>	char*
<i>jump_↔ instruction_↔ number</i>	int

## Returns

void None

4.1.3.5 void gencode ( const char \* *result*, const char \* *addr1* )

Generate 3 address code and store in Quadrule table for assignement e.g.

a = t0;

## Parameters

<i>result</i>	char*
<i>address2</i>	char*

## Returns

void None

4.1.3.6 void gencode ( const char \* *result*, int *label* )

Generate 3 address code and store in Quadrule table for goto target e.g.

'goto' -1

## Parameters

<i>goto</i>	char*
<i>jump_</i> ↔ <i>instruction_</i> ↔ <i>number</i>	int

## Returns

void None



# Index

info, [6](#)

marks, [7](#)

next, [7](#)

Quadruple, [8](#)