

1. Copyright.

Copyright © Dave Bone 1998 - 2015

2. *prefile_include* Thread.

File name recognizer for includes. It checks the file system to verify that the file exists. Inside the returned “file-inclusion” terminal is the file open findings whether the file name exists. It is up to the caller to check this error status. A null status means it’s okay while “bad filename” error symbol with the extracted filename string is used.

3. Fsm *Cprefile_include* class.**4. *Cprefile_include* constructor directive.**

```
<Cprefile_include constructor directive 4> ≡
  chrs_ = 0;
  error_raised_ = 0;
```

5. *Cprefile_include* op directive.

```
<Cprefile_include op directive 5> ≡
  chrs_ = 0;
  error_raised_ = 0;
```

6. *Cprefile_include* user-declaration directive.

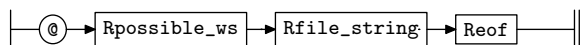
```
<Cprefile_include user-declaration directive 6> ≡
public: T_c_string * chrs_;
  yacco2 :: CAbs_lr1_sym * error_raised_;
```

7. *Cprefile_include* user-prefix-declaration directive.

```
<Cprefile_include user-prefix-declaration directive 7> ≡
#include "ws.h"
#include "eol.h"
#include "c_string.h"
  using namespace NS_yacco2_terminals;
```

8. *Rprefile_include* rule.

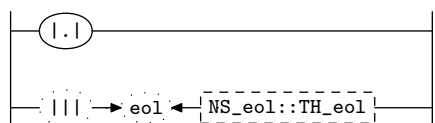
Rprefile_include

**9. *Rprefile_include* op directive.**

```
<Rprefile_include op directive 9> ≡
  Cprefile_include * fsm = ( Cprefile_include * ) rule_info_...parser_...fsm_tbl_;
  T_file_inclusion * typ = new T_file_inclusion(fsm->chrs_, fsm->error_raised_);
  typ->set_rc(*rule_info_...parser_...start_token_, __FILE__, __LINE__);
  RSVP(typ);
  fsm->chrs_ = 0;
```

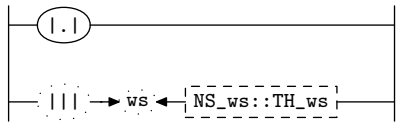
10. *Reof* rule.

Reof



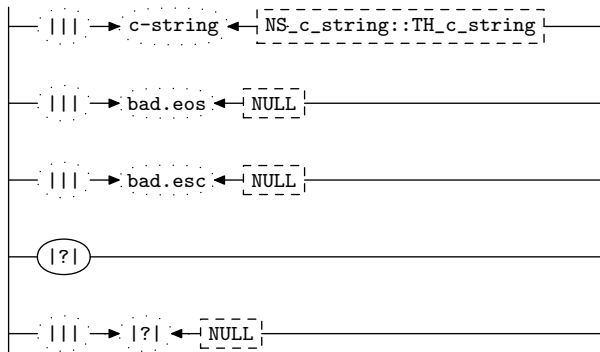
11. *Rpossible_ws* rule.

Rpossible_ws

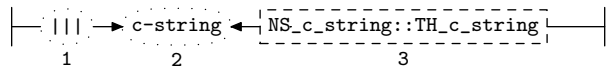


12. *Rfile_string* rule.

Rfile_string



13. *Rfile_string*'s subrule 1.



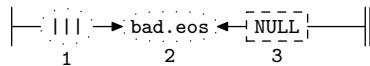
$\langle \text{Rfile_string subrule 1 op directive 13} \rangle \equiv$

```

Cprefile_include * fsm = ( Cprefile_include * ) rule_info_.parser--fsm_tbl_;
fsm->chrs_ = sf-p2--;
string * ps_fn = fsm->chrs->c_string();
std::ifstream fle_chk;
fle_chk.open(ps_fn->c_str());
if (!fle_chk.good()) {
    fsm->error_raised_ = new Err_bad_filename(*ps_fn);
    fsm->error_raised_-set_rc(*rule_info_.parser--start_token_, __FILE__, __LINE__);
}
else {
    fle_chk.close();
}

```

14. *Rfile_string*'s subrule 2.

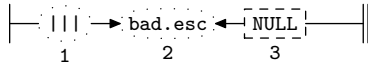


$\langle \text{Rfile_string subrule 2 op directive 14} \rangle \equiv$

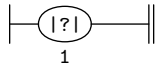
```

Cprefile_include * fsm = ( Cprefile_include * ) rule_info_.parser--fsm_tbl_;
fsm->error_raised_ = sf-p2--;

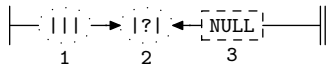
```

15. *Rfile_string*'s subrule 3.

⟨Rfile_string subrule 3 op directive 15⟩ ≡
Cpreinclude * *fsm* = (*Cpreinclude* *) *rule_info...**parser--fsm_tbl_*;
fsm-error_raised_ = *sf-p2_*;

16. *Rfile_string*'s subrule 4.

⟨Rfile_string subrule 4 op directive 16⟩ ≡
Cpreinclude * *fsm* = (*Cpreinclude* *) *rule_info...**parser--fsm_tbl_*;
fsm-error_raised_ = **new** *Err_no_filename*;
fsm-error_raised_→*set_rc*(**rule_info...**parser--current_token*(), *__FILE__*, *__LINE__*);

17. *Rfile_string*'s subrule 5.

⟨Rfile_string subrule 5 op directive 17⟩ ≡
Cpreinclude * *fsm* = (*Cpreinclude* *) *rule_info...**parser--fsm_tbl_*;
fsm-error_raised_ = *sf-p2_*;

18. First Set Language for O_2^{linker} .

```
/*
  File: prefile_include.fsc
  Date and Time: Fri Jan  2 15:33:49 2015
*/
transitive      n
grammar-name    "prefile_include"
name-space     "NS_prefile_include"
thread-name     "TH_prefile_include"
monolithic     n
file-name      "prefile_include.fsc"
no-of-T        569
list-of-native-first-set-terminals 1
  raw_at_sign
end-list-of-native-first-set-terminals
list-of-transitive-threads 0
end-list-of-transitive-threads
list-of-used-threads 3
  NS_c_string::TH_c_string
  NS_eol::TH_eol
  NS_ws::TH_ws
end-list-of-used-threads
fsm-comments
"Preprocessor source file from the 'include file' directive."
```

19. Lr1 State Network.

\Rightarrow		State: 1 state type: s	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
c	Rpreprofile_include	1 1 1 @	1 2 8
\Rightarrow	@	State: 2 state type: s	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
c	Rpossible_ws	3 2 1 ws NS_ws::TH_ws	2 9 10
c	Rpossible_ws	3 1 1 .	2 11 11
t	Rpreprofile_include	1 1 2 Rpossible_ws <u>Rfile_string</u>	1 3 8
\Rightarrow	Rpossible_ws	State: 3 state type: s	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
c	Rfile_string	4 4 1 ?	3 12 12
c	Rfile_string	4 2 1 bad eos NULL	3 13 16
c	Rfile_string	4 5 1 ? NULL	3 13 14
c	Rfile_string	4 1 1 c-string NS_c_string::TH_c_string	3 13 15
c	Rfile_string	4 3 1 bad esc NULL	3 13 17
t	Rpreprofile_include	1 1 3 Rfile_string <u>Reof</u>	1 4 8
\Rightarrow	Rfile_string	State: 4 state type: s	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
c	Reof	2 2 1 eol NS_eol::TH_eol	4 5 6
c	Reof	2 1 1 .	4 7 7
t	Rpreprofile_include	1 1 4 Reof	1 8 8
\Rightarrow	arbitration-code: ϵ	State: 5 state type: s	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
t	Reof	2 2 2 eol	4 6 6
\Rightarrow	eol	State: 6 state type: r	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
t	Reof	2 2 3	4 0 6 1
\Rightarrow	.	State: 7 state type: r	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
t	Reof	2 1 2	4 0 7 1
\Rightarrow	Reof	State: 8 state type: r	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
t	Rpreprofile_include	1 1 5	1 0 8 1
\Rightarrow	arbitration-code: ϵ	State: 9 state type: s	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
t	Rpossible_ws	3 2 2 ws	2 10 10
\Rightarrow	ws	State: 10 state type: r	
\leftarrow	rule	\rightarrow R# sr# Po \leftarrow	subrule element \rightarrow Brn Gto Red LA
t	Rpossible_ws	3 2 3	2 0 10 2
\Rightarrow	.	State: 11 state type: r	

<p>← rule → R# sr# Po ← t Rpossible_ws 3 1 2</p>	<p>subrule element</p>	<p>→ Brn Gto Red LA 2 0 11 2</p>
⇒ ?		
<p>← rule → R# sr# Po ← t Rfile_string 4 4 2</p>	<p>State: 12 state type: <i>r</i> subrule element</p>	<p>→ Brn Gto Red LA 3 0 12 3</p>
⇒ <i>arbitration-code: ε</i>		
<p>← rule → R# sr# Po ← t Rfile_string 4 5 2 ? t Rfile_string 4 1 2 c-string t Rfile_string 4 2 2 bad eos t Rfile_string 4 3 2 bad esc</p>	<p>State: 13 state type: <i>s</i> subrule element</p>	<p>→ Brn Gto Red LA 3 14 14 3 15 15 3 16 16 3 17 17</p>
⇒ ?		
<p>← rule → R# sr# Po ← t Rfile_string 4 5 3</p>	<p>State: 14 state type: <i>r</i> subrule element</p>	<p>→ Brn Gto Red LA 3 0 14 3</p>
⇒ <i>c-string</i>		
<p>← rule → R# sr# Po ← t Rfile_string 4 1 3</p>	<p>State: 15 state type: <i>r</i> subrule element</p>	<p>→ Brn Gto Red LA 3 0 15 3</p>
⇒ <i>badeos</i>		
<p>← rule → R# sr# Po ← t Rfile_string 4 2 3</p>	<p>State: 16 state type: <i>r</i> subrule element</p>	<p>→ Brn Gto Red LA 3 0 16 3</p>
⇒ <i>badesc</i>		
<p>← rule → R# sr# Po ← t Rfile_string 4 3 3</p>	<p>State: 17 state type: <i>r</i> subrule element</p>	<p>→ Brn Gto Red LA 3 0 17 3</p>

20. Index.

`|.`: 10, 11.
`||`: 10, 11, 12.
`|?`: 12.
`__FILE__`: 9, 13, 16.
`__LINE__`: 9, 13, 16.
`bad eos`: 12.
`bad esc`: 12.
`c-string`: 12.
`c_str`: 13.
`c_string`: 13.
`CAbs_lr1_sym`: 6.
`chrs`: 4, 5, 6, 9, 13.
`close`: 13.
`Cprefile_include`: 9, 13, 14, 15, 16, 17.
`current_token`: 16.
`eol`: 10.
`Err_bad_filename`: 13.
`Err_no_filename`: 16.
`error_raised`: 4, 5, 6, 9, 13, 14, 15, 16, 17.
`fle_chk`: 13.
`fsm`: 9, 13, 14, 15, 16, 17.
`fsm_tbl`: 9, 13, 14, 15, 16, 17.
`good`: 13.
`ifstream`: 13.
`NS_c_string::TH_c_string`: 12.
`NS_eol::TH_eol`: 10.
`NS_ws::TH_ws`: 11.
NS_yacco2_terminals: 7.
`NULL`: 12.
`open`: 13.
`parser`: 9, 13, 14, 15, 16, 17.
`prefile_include`: 2.
`ps_fn`: 13.
`p2`: 13, 14, 15, 17.
`Reof`: 8.
`Reof`: 10.
`Rfile_string`: 8.
`Rfile_string`: 12, 13, 14, 15, 16, 17.
`Rpossible_ws`: 8.
`Rpossible_ws`: 11.
`Rprefile_include`: 8.
`RSVP`: 9.
`rule_info`: 9, 13, 14, 15, 16, 17.
`set_rc`: 9, 13, 16.
`sf`: 13, 14, 15, 17.
`start_token`: 9, 13.
`std`: 13.
`string`: 13.
`T_c_string`: 6.
`T_file_inclusion`: 9.
`typ`: 9.
`ws`: 11.
`yacco2`: 6.

⟨ Cprefile_include constructor directive 4 ⟩
⟨ Cprefile_include op directive 5 ⟩
⟨ Cprefile_include user-declaration directive 6 ⟩
⟨ Cprefile_include user-prefix-declaration directive 7 ⟩
⟨ Rfile_string subrule 1 op directive 13 ⟩
⟨ Rfile_string subrule 2 op directive 14 ⟩
⟨ Rfile_string subrule 3 op directive 15 ⟩
⟨ Rfile_string subrule 4 op directive 16 ⟩
⟨ Rfile_string subrule 5 op directive 17 ⟩
⟨ Rprefile_include op directive 9 ⟩

preprofile_include Grammar

Date: January 2, 2015 at 15:38

File: preprofile_include.lex Ns: NS_preprofile_include

Version: 1.0

Debug: false

Grammar Comments:

Type: Thread

Preprocessor source file from the "include file" directive.

1 element(s) in Lookahead Expression below

eor

	Section	Page
Copyright	1	1
<i>preprofile_include</i> Thread	2	2
Fsm Cpreprofile_include class	3	2
Cpreprofile_include constructor directive	4	2
Cpreprofile_include op directive	5	2
Cpreprofile_include user-declaration directive	6	2
Cpreprofile_include user-prefix-declaration directive	7	2
<i>Rpreprofile_include</i> rule	8	2
Rpreprofile_include op directive	9	2
<i>Reof</i> rule	10	2
<i>Rpossible_ws</i> rule	11	3
<i>Rfile_string</i> rule	12	3
<i>Rfile_string</i> 's subrule 1	13	3
<i>Rfile_string</i> 's subrule 2	14	3
<i>Rfile_string</i> 's subrule 3	15	4
<i>Rfile_string</i> 's subrule 4	16	4
<i>Rfile_string</i> 's subrule 5	17	4
First Set Language for O_2^{linker}	18	5
Lr1 State Network	19	6
Index	20	8